WORLDAUDITBENCH: INTERACTIVE 3D WORLD AUDITING WITH MULTIMODAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

As interactive 3D worlds are increasingly used to study intelligent behavior, it becomes important to develop efficient pipelines for identifying anomalies in these simulated environments, such as floating objects, traversable walls, or objects inconsistent with the surrounding scene. Multimodal AI systems, including vision-language models (VLMs) and vision-language-action models (VLAs), have shown potential for automating this task. However, 3D world auditing is complex, requiring the close coupling of two distinct capabilities: **action**, to navigate the 3D world and search for anomalies systematically and efficiently; and **visual reasoning**, to understand the environment and identify anomalies from multimodal observations. It remains largely unexplored whether multimodal agents can effectively couple these two capabilities—using visual reasoning to identify potential anomalies while taking actions to validate them. In this paper, we introduce **WorldAuditBench**, a benchmark for 3D world auditing comprising 213 anomaly tasks across 13 environments built with Unreal Engine 5 and Three.js, spanning five anomaly families. We evaluate five frontier models under a fixed exploration budget using two auditing paradigms: VLA-based exploration followed by VLM-based anomaly identification, and an end-to-end VLM agent in which visual reasoning directly guides action selection. Across the evaluated models and two paradigms, success rates range from 6.6% to 42.3%, substantially below human performance (83.4%). Through the task of world auditing, WorldAuditBench provides a testbed for studying how multimodal agents couple action and visual reasoning in interactive 3D environments, while highlighting current limitations in their ability to gather and interpret evidence during exploration. Project page: `https://work50659.github.io/demov1/`.

1 INTRODUCTION

Interactive 3D worlds allow human players and agents to explore, interact with objects, and complete tasks in realistic simulated environments. Recent work has expanded their use in embodied AI, from navigation and manipulation to open-ended exploration (Yang et al., 2024b; Xie et al., 2025; Lee et al., 2025; Yang et al., 2025b; Ye et al., 2025). Yet a world can look convincing while still containing anomalies. For example, a chair may float above the floor, a solid wall may allow an agent to walk through it, or an object may disappear after the agent looks away. Recent works highlight such visual and physical defects, while others develop methods to improve object placement, collision handling, and physical consistency (Yang et al., 2024a; Zhou et al., 2026; Jia & Chen, 2025; Lin et al., 2025; Taesiri et al., 2025). Such anomalies matter wherever interactive worlds are used to study intelligent behavior. For example, an agent that succeeds at certain tasks by walking through a broken wall should not be seen as reliable in navigation. Checking these environments therefore requires more than inspecting how they look, since some defects become apparent only through interaction or repeated exploration in the environments. This raises a natural question: *Can multimodal agents autonomously identify anomalies in a 3D world, i.e., perform **world auditing**?*

World auditing requires two skills: **action**, to navigate the 3D world and search for anomalies systematically and efficiently, and **visual reasoning**, to understand the scene and identify anomalies from observations. More importantly, these two skills are deeply coupled. The agent must search large spaces within a limited exploration budget while distinguishing real defects from unusual but

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



Figure 1: **Anomaly taxonomy of WorldAuditBench (Section 3.2).** Representative cases illustrate five families of physical, spatial, temporal, and semantic inconsistencies.

valid appearances or behavior. An observation may reveal a possible anomaly without providing enough evidence to confirm it. The agent must then choose an action to further investigate and decide whether a real defect is present. Auditing therefore requires both skills to work together: visual reasoning guides what to test, and action provides further evidence for the diagnosis. For example, the fence in Figure 1 *Missing Collision Anomaly* appears to be a solid barrier, but its collision defect becomes apparent only through interaction. The agent must navigate toward the fence and attempt to cross it. Comparing views in the same direction before and after moving straight ahead by 3 m reveals that the agent passed through the fence without being blocked. This process connects visual reasoning and action: recognizing the barrier establishes an expectation about blocked movement, moving forward tests it, and the resulting observation provides evidence of the collision failure.

Existing benchmarks evaluate visual reasoning over fixed observations or interactive task completion (Taesiri et al., 2025; Yang et al., 2025a; Majumdar et al., 2024), while interactive anomaly search covers out-of-place objects (Pardyl et al., 2025). Comprehensive evaluation of world auditing remains underexplored: agents must act to gather evidence and identify diverse anomalies, including those revealed only through interaction or comparison across viewpoints and time.

To fill this gap, we introduce WorldAuditBench, comprising 213 anomaly tasks across 13 environments implemented in Unreal Engine and Three.js. As illustrated in Figure 1, these environments cover indoor, city, historical, industrial, and natural settings. We organize anomalies into five families: static physics, interactive physics, spatial consistency, temporal consistency, and semantic consistency. Together, these families cover both visible defects and anomalies that require interaction or repeated observation to detect. Each task pairs an anomaly with its expected normal behavior, relevant scene context, and an evaluation rubric. To solve a task, the auditor explores the scene from a first-person perspective and submits a report describing the anomaly and supporting observations. Table 1 compares WorldAuditBench with prior benchmarks and simulation environments, highlighting its focus on interactive anomaly identification across diverse anomaly families.

To study how action and visual reasoning work together in world auditing, we evaluate two auditor setups: ① a *single VLM agent auditor*, which uses a single multimodal LLM to inspect observations, select actions, and report anomalies; and ② a *two-stage VLA–VLM agent auditor*, which uses a VLA model to explore the environment and then passes the resulting trajectory to a VLM for offline anomaly analysis. The key difference is whether anomaly analysis can guide exploration process: the VLM agent can adjust its actions to investigate suspected anomalies, while the two-stage auditor analyzes a fixed trajectory, resembling a conventional video QA setup.

Table 1: **Comparison with related benchmarks and simulation environments.** WorldAuditBench evaluates interactive anomaly identification across 15 categories grouped into five families.

Work	Task family	World interaction	Anomaly identification	Anomaly coverage
Offline visual benchmarks				
Video-MME (2025)	Question answering	✗	✗	—
GlitchBench (2024)	Game testing	✗	✓	Game glitches
VideoGameQA-Bench (2025)	Game testing	✗	✓	Game glitches
VideoGlitchBench (2026)	Game testing	✗	✓	Game glitches
Interactive environments and benchmarks				
SimWorld (2025)	Task execution	✓	✗	—
UnrealZoo (2025)	Task execution	✓	✗	—
EmbodiedBench (2025a)	Task execution	✓	✗	—
OpenEQA (active) (2024)	Question answering	✓	✗	—
FlySearch (2025)	Target search	✓	✓	Out-of-place objects
WorldAuditBench (ours)	World auditing	✓	✓	15 categories (§3.2)

Across the 213 evaluation tasks, *VLM agents* achieve success rates of 28.2% to 42.3%, compared with 6.6% to 17.4% for *two-stage VLA–VLM auditors*. These results suggest that using observations to guide further exploration helps agents gather evidence to verify suspected defects. However, both approaches fall well below the human success rate of 83.4%, leaving a substantial gap in their ability to audit interactive worlds.

To better understand this gap, we evaluate five model backbones and conduct controlled ablations on environment initialization and in-context examples. Further analysis identifies difficulties in both exploration and the use of past observations. The evaluated VLA models struggle to reach anomalies in large 3D worlds, limiting the evidence available for subsequent analysis. For VLM agents, the memory tool that lets them store and retrieve observations during exploration substantially improves anomaly discovery. These findings suggest that world auditing benefits from both exploration guided by ongoing analysis and access to evidence gathered earlier in the trajectory.

Our contributions are listed below:

- **A comprehensive benchmark for interactive world auditing.** We introduce WorldAuditBench, with 213 tasks across 13 environments, covering static physics, interactive physics, spatial consistency, temporal consistency, and semantic consistency. These tasks evaluate anomaly identification through visual inspection, interaction, and repeated observation.
- **A detailed analysis of world auditing agents.** We compare VLM agents with two-stage VLA–VLM auditors to examine their exploration and reasoning abilities. Our analysis further highlights exploration failures in VLA models and the benefits of memory control for VLM agents, while documenting a substantial performance gap relative to human auditors.

2 Related Work

Visual Understanding and Anomaly Detection. Existing works mainly evaluate agents' multimodal abilities using static visual assets. For example, Video-MME focuses on prerecorded videos (Fu et al., 2025), while VideoGameBunny studies game video clips (Taesiri & Bezemer, 2025). GlitchBench, VideoGameQA-Bench, and VideoGlitchBench further evaluate glitch recognition, localization, description, and reporting from images or videos (Taesiri et al., 2024; 2025; Zheng et al., 2026). However, these benchmarks provide observations to the model in advance and mainly test whether agents can answer questions reliably based on available visual evidence. In contrast, our work focuses on the more challenging setting of interactive world auditing, where agents must actively gather the evidence needed to expose and verify anomalies.

Interactive Embodied Reasoning and Anomaly Search. Existing works also evaluate agents' embodied abilities in interactive environments. For example, AI2-THOR (Kolve et al., 2017), Habitat (Savva et al., 2019), UnrealCV (Qiu & Yuille, 2016), SimWorld (Ren et al., 2025), and UnrealZoo (Zhong et al., 2025) provide 3D environments for embodied perception and interaction, while

EmbodiedBench (Yang et al., 2025a), VisualAgentBench (Liu et al., 2024), and OpenEQA (Majumdar et al., 2024) evaluate agents on interactive task execution and information gathering. More closely related to our work, FlySearch (Pardyl et al., 2025) includes an anomaly-search task that requires agents to locate out-of-place objects in 3D environments. However, they mainly focus on task completion or locating visually identifiable anomalies. In contrast, our work studies anomalies that may require agents to actively test physical properties or compare observations across viewpoints and time, making evidence gathering itself an essential part of anomaly identification.

3 WORLDAUDITBENCH DATASET

This section introduces WorldAuditBench and its construction. Section 3.1 provides an overview, Section 3.2 presents our anomaly taxonomy, and Section 3.3 describes the data collection process.

3.1 DATASET OVERVIEW

WorldAuditBench consists of 213 tasks built on 13 interactive 3D environments, each task implanted with a **single** pre-defined, labeled anomaly. An agent explores and interacts with the environment, and its goal is to correctly identify the anomaly based on its observations. The environments span five scene types (indoor, urban, historical, industrial, and natural) and are built with Unreal Engine 5[1] (126 tasks) or Three.js[2] (87 tasks). Appendix A provides further details.

3.2 ANOMALY TAXONOMY

Our anomaly taxonomy draws on game-bug studies that distinguish failures visible in a single state from those requiring observations over time (Lewis et al., 2010), and describe failures in object position, collision, rendering, persistence, and state transitions (Truelove et al., 2021; Butt et al., 2023). We group anomalies by their observable effects into five families (Figure 1):

- **Static physics:** Anomalies in which static objects disobey physics, involving object support, overlap, or scale;
- **Interactive physics:** Anomalies in which interactions between objects or between the agent and the environment disobey physics, involving obstruction and objects' responses to contact;
- **Spatial consistency:** Anomalies in which objects are inconsistent across positions or viewpoints, such as appearance changes in an object as the observer moves;
- **Temporal consistency:** Anomalies in which objects display unexpected changes over time, such as changes in existence, attributes, and behavior;
- **Semantic consistency:** Anomalies in which object configurations are inconsistent with the semantic context, involving the intended functionality of the object and historical context.

The five families of anomaly are further divided into fifteen categories. Appendix C.5 provides full definitions of each category and some examples. Anomaly families differ in the evidence required to detect them: some are visible in a single view, such as those in the static physics family, while others can only be exposed through deliberate actions, such as interacting with objects, changing viewpoints, or revisiting a location after a period of time. Together, these families form a comprehensive test of planning, memory, and scene understanding.

3.3 TASK CONSTRUCTION

To construct tasks based on the aforementioned taxonomy, we follow three steps: scene collection, task construction, and human validation (Figure 2).

Scene Collection. We collect 13 interactive 3D environments built with Unreal Engine 5 and Three.js from publicly available sources, covering indoor, urban, historical, industrial, and natural settings. These environments range from furnished rooms and urban streets to historical markets and outdoor landscapes. For larger environments, we manually select several bounded regions as individual scenes, seeking distinct spatial layouts and scene content, yielding 27 scenes in total. Appendix A presents the environments and representative views.

[1] https://www.unrealengine.com
[2] https://threejs.org



Figure 2: **Task construction in WorldAuditBench.** We collect scenes, introduce anomalies, and iteratively refine tasks through human review of their quality.

Task Construction. All tasks are hand-crafted by the authors. To construct a task, we first select a bug-free scene, where we further define permissible regions for exploration, the starting position and orientation of the agent, and available interactions. Based on the taxonomy, we then introduce an anomaly by perturbing the scene, such as changing object placement, collision, appearance, responses to interaction, or behavior over time. For semantic anomalies, we rearrange objects in ways that conflict with their intended function or add objects that do not fit the scene's historical setting. Appendix B.1 details the construction procedure. Each task is annotated with the ground-truth anomaly category along with an evaluation rubric describing the target anomaly and its expected normal appearance or behavior. Appendix B.2 presents an example.

Human Validation. To verify task quality, judges who did not participate in the task construction process, are asked to explore the scene in each task to check whether the target anomaly can be observed and verified and whether it matches the annotations. They rate task quality as Pass, Fail, or Uncertain. Fail or Uncertain judgments lead to further revision and review. This process repeats until the task meets the acceptance requirement: *Rated Pass by at least **two judges** on the same task version*. Appendix B.3 describes the review protocol and annotation interface.

4 Auditing Paradigms

To study how the coupling of **action** and **visual reasoning** affects 3D world auditing, we evaluate two paradigms: VLM-only and VLA–VLM (Figure 3). We first define the task for the auditing agent in Section 4.1, and then describe these two paradigms in Sections 4.2 and 4.3.

4.1 Task Definition

At the start of a task, the auditor receives an initial input x containing auditing instructions, task context, and an initial observation. The task context includes a scene description, an anomaly-type hint, and a matching in-context example. The initial observation is the first-person RGB view from the starting pose. From this input, the auditor explores the environment under a fixed budget and produces a final anomaly report containing the identified anomaly and supporting observation images during its exploration and interaction within the 3D world. Each task in the main evaluation contains *only one* target anomaly, and success is determined by whether the report identifies it. In both paradigms, the agent submits an anomaly report and evidence to the VLM judge, which evaluates them against the task's evaluation rubric, such as identifying a solid wall that the agent can pass through. Appendices C.3 and C.4 illustrate an example report and its evaluation process.

To formalize this interaction, we represent the trajectory of an auditor agent as

$$\tau = (x, a_0, o_1, \ldots, a_{T-1}, o_T, y), \tag{1}$$



Figure 3: **Two auditing paradigms.** (a) VLM agents reason and use tools during exploration. (b) VLA–VLM auditors analyze trajectories after VLA exploration. Both use the same judge.

where a_t is an environment action, o_{t+1} is the resulting observation, and y is the final anomaly report. As with the initial observation, each subsequent observation contains a first-person RGB view and available interaction hints, *e.g.*, that the agent can open or close a door. The auditor selects each action based on the initial input and the preceding actions and observations.

4.2 VLM-Only Paradigm

In the VLM-only paradigm, a VLM auditor receives the initial input x (Appendix C.1). At step t, it reasons over the trajectory collected so far, including the current observation o_t, to choose the next environment action a_t. A suspected anomaly can therefore guide where the agent moves, which viewpoint it examines, or which interaction it attempts. Visual reasoning and action remain coupled as the trajectory is collected.

The agent uses **environment actions** to navigate, change its view, interact with objects, and observe changes over time; **memory tools** to retrieve earlier frames, review history, and save notes; and **anomaly-reporting tools** to record or revise findings and link supporting observations. Figure 3(a) illustrates revisiting a roadside location, retrieving an earlier observation, and reporting a missing sign using the matched before-and-after views as evidence. Tool definitions are shown in Table 6.

4.3 VLA–VLM Paradigm

The VLA–VLM paradigm separates exploration from anomaly analysis (Figure 3(b)). A VLA model first generates environment actions a_t and collects a trajectory of observations. Due to the limitations of existing VLA models, no explicit reasoning is produced during exploration. A VLM then analyzes the recorded observations together with the task context to identify anomalies and produce a report with supporting evidence. In our experiments, we use fixed recorded trajectories recorded by the same VLA model for second-stage analysis, which is conducted by different VLM backbones. Because the VLM analyzes the trajectory only after exploration is complete, its intermediate observations cannot guide further actions to investigate potential anomalies, which may limit its auditing performance.

Table 2: **Auditing performance and cost on WorldAuditBench.** SR (%) and cost ($/task). SP: static physics; IP: interactive physics; SpC: spatial consistency; TC: temporal consistency; SeC: semantic consistency. **Bold** and underline mark the best and second-best results.

Auditor	SP	IP	SpC	TC	SeC	**Overall ↑**	Cost ↓
Human	89.8	81.3	76.3	78.1	88.6	83.4	N/A
VLM auditors							
GPT-6 Astra	**59.3**	**29.3**	**45.1**	**12.5**	**68.2**	**42.3**	2.955
Claude Opus 5	35.6	19.5	29.4	**12.5**	50.0	28.2	2.445
Gemini 3.8 Flash	49.2	22.0	29.4	**12.5**	50.0	32.4	1.382
Muse Spark 1.3	22.0	12.2	13.7	5.0	22.7	15.0	0.791
Qwen 3.8 Flash	11.9	7.3	3.9	2.5	22.7	8.5	**0.077**
VLA–VLM auditors							
GPT-6 Astra	20.3	17.1	**25.5**	0.0	13.6	16.4	0.281
Claude Opus 5	**22.0**	22.0	11.8	2.5	13.6	15.0	0.727
Gemini 3.8 Flash	20.3	**24.4**	17.6	**5.0**	**18.2**	**17.4**	0.296
Muse Spark 1.3	5.1	9.8	9.8	0.0	9.1	6.6	0.112
Qwen 3.8 Flash	18.6	22.0	11.8	0.0	4.5	12.7	**0.039**

5 Experiments

5.1 Experimental Setup and Metrics

Agent Implementation. We evaluate GPT-6 Astra, Claude Opus 5, Gemini 3.8 Flash, Muse Spark 1.3, and Qwen 3.8 Flash under the two auditing paradigms in Section 4. We run GPT-6 Astra with Codex CLI (OpenAI, 2026), Claude Opus 5 with Claude Code (Anthropic, 2026), Gemini 3.8 Flash with Gemini CLI (Google, 2026), Muse Spark 1.3 with OpenCode (Anomaly, 2026), and Qwen 3.8 Flash with Qwen Code (Qwen Team, 2026). These harnesses manage model calls and conversation history, while a shared Model Context Protocol (MCP) interface (Model Context Protocol, 2025) exposes the auditing tools in Table 6. Each comparison thus evaluates a model together with its harness. In the VLM-only paradigm, VLM auditors receive a budget of 40 environment actions. In the VLA–VLM paradigm, we use Open-P2P 1.2B (Yue et al., 2026) as the VLA model to collect 60 seconds of simulated exploration, with observations sampled every 0.5 seconds. Recorded trajectories are shared across VLM backbones; the two paradigms use different exploration budgets.

Human Baseline. Besides AI agents, we evaluate human performance on this task. Our human evaluation involves approximately ten computer science PhD students. Each task is evaluated by two participants. Participants receive the same task information as the models, including the scene description, auditing instructions, and in-context examples (Appendix C.1). They have up to ten minutes per task to explore the environment and report their findings. Human reports are evaluated by the same VLM judge against the same evaluation rubrics as model reports.

Metrics. Following the VLM-as-a-judge paradigm, we use GPT-6 Astra to assess each report and its evidence against the task's evaluation rubric. We report success rate (SR), the percentage of tasks for which the target anomaly is correctly identified. Appendix C.4 provides the judge prompt and an example evaluation. Table 7 reports agreement between the VLM judge and human judgments.

5.2 Main Results

Table 2 shows that auditing remains challenging across five backbones. Among VLM auditors, GPT-6 Astra achieves the highest SR at 42.3%, followed by Gemini at 32.4% and Claude at 28.2%, while Muse and Qwen reach 15.0% and 8.5%. Across the five anomaly families, static physics and semantic consistency are relatively easy for VLM auditors, as the relevant physical or contextual inconsistency can often be identified from a single frame. Spatial and temporal consistency require comparing observations across viewpoints or time, and receive lower scores. Temporal consistency is particularly challenging, with SR at or below 12.5% for every model and paradigm. Appendix F illustrates difficulties in gathering before-and-after evidence and identifying the target change.

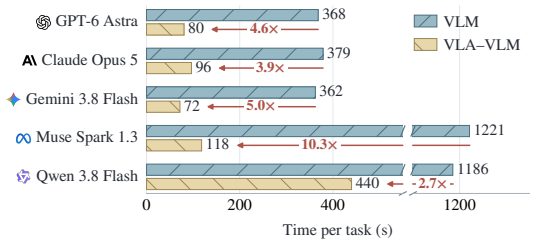


Figure 4: **Auditing time per task.**

Most models achieve higher SR with interactive VLM auditing than with VLA–VLM analysis of fixed trajectories. Interactive auditors can choose additional viewpoints and actions as they reason, whereas VLA–VLM auditors analyze fixed trajectories. Section 6.3 examines anomaly reach and recognition under these evidence-gathering procedures. Figure 4 compares the auditing time of the two paradigms. Given prerecorded trajectories, VLA–VLM analysis is 2.7–10.3× faster than interactive VLM auditing and has lower inference costs, these efficiency gains come at the cost of lower SR, reflecting a trade-off between efficient trajectory analysis and adaptive evidence gathering.

6 Ablation Studies

6.1 Factors Affecting Auditing Performance

We study how starting distance, exploration budget, and task guidance affect auditing with Gemini 3.8 Flash on 126 Unreal tasks (Table 3).

Table 3: **Auditing factors.** SR (%), Δ (pp).

Setting	VLM	VLA–VLM
Default	**33.3**	**5.6**
(a) Starting distance		
Near (1/3)	38.1 (+4.8)	7.1 (+1.6)
(b) Exploration budget		
0.5×	23.0 (−10.3)	4.0 (−1.6)
1.5×	34.9 (+1.6)	7.1 (+1.6)
(c) Task guidance		
Without ICL	31.0 (−2.4)	5.6 (0.0)
Without ICL & type hint	15.9 (−17.5)	2.4 (−3.2)

Starting Distance. We move the starting position closer to the target anomaly, reducing the navigable route distance to approximately one third of its original length while keeping the agent's initial orientation unchanged. Moving closer increases the VLM-only success rate from 33.3% to 38.1% and the VLA–VLM success rate from 5.6% to 7.1%. These results indicate that auditing becomes more challenging when agents start farther from the anomaly.

Exploration Budget. We express exploration budgets relative to each approach's default (1×): 40 steps for VLM and 60 seconds for VLA–VLM. The 0.5× and 1.5× settings correspond to 20 and 60 steps for VLM, and 30 and 90 seconds for VLA–VLM. At 0.5×, success rate drops from 33.3% to 23.0% for VLM and from 5.6% to 4.0% for VLA–VLM. At 1.5×, success rate increases to 34.9% for VLM and 7.1% for VLA–VLM.

Task Guidance. By default, agents receive an anomaly-type hint and a matching in-context example. We first remove thein-context example while retaining the anomaly-type hint. VLM success rate decreases from 33.3% to 31.0%, while VLA–VLM success rate remains at 5.6%. Further removing the type hint reduces success rates to 15.9% for VLM and 2.4% for VLA–VLM. The anomaly-type hint matters far more than the in-context example, suggesting that knowing what to look for is critical for directing exploration.

6.2 Auditing Multiple Anomalies In a Scene

To test auditing with coexisting anomalies, we select 21 anchor tasks, three from each of seven Unreal Engine 5 environments, and evaluate the Gemini 3.8 Flash VLM auditor with a 40-step budget. The original anomaly is called the *anchor anomaly*. For each anchor, we randomly introduce one or two additional anomalies on distinct objects to form two- and three-anomaly conditions, sampling two configurations per anchor for each condition, yielding 42 runs per condition. Agents are given the total number of anomalies, but only the anchor anomaly's type and in-context example. We report the success rate in identifying the anchor anomaly (anchor SR), the success rate in identifying every anomaly (all-anomaly SR), and the fraction of anomalies identified (target recall).

Figure 5 shows that anchor SR remains similar across anomaly counts: 42.9% with one or two anomalies and 47.6% with three. Complete discovery remains difficult: the agent identifies every target in only 3/42 two-anomaly runs (7.1%) and 2/42 three-anomaly runs (4.8%), with target recall

of 29.8% and 29.4%, respectively. Under a fixed exploration budget, identifying the anchor often leaves other coexisting anomalies undiscovered.

6.3 Anomaly Exposure and Identification

As discussed in Section 4, VLM auditing couples action with visual reasoning. Using GPT-6 Astra on 213 tasks, we examine two aspects of its advantage over VLA–VLM auditing: ❶ **anomaly exposure**, whether exploration reveals the anomaly, and ❷ **anomaly identification upon exposure**, whether the agent identifies the anomaly once it is revealed. We assess exposure using geometric coverage, based on the agent's position and orientation, and human-rated exposure, based on exploration videos. For identification, we compare success rates on exposed tasks under both measures.

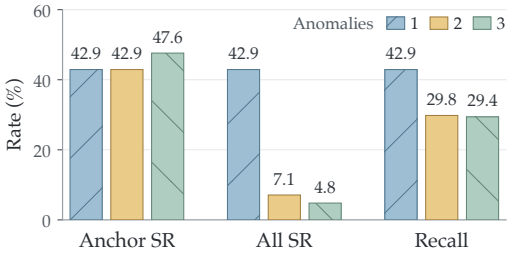


Figure 5: **Auditing multiple anomalies.**

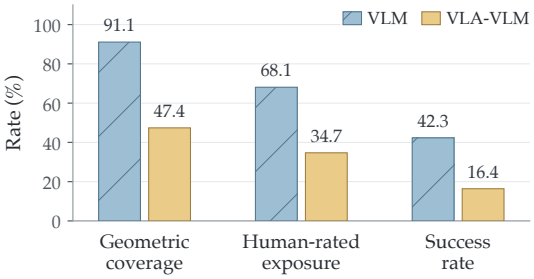


Figure 6: **Anomaly exposure and success.**

Anomaly Exposure. Geometric coverage is 91.1% for the VLM agent and 47.4% for the VLA explorer, while human-rated exposure is 68.1% and 34.7%, respectively (Figure 6). For spatial and temporal consistency anomalies, human-rated exposure requires contrasting viewpoints or before-and-after states, whereas geometric coverage requires only one qualifying frame. The higher exposure of the VLM agent suggests that reasoning and planning help direct exploration toward relevant parts of the scene. Given in-context examples, the VLM agent can reason about what an anomaly may look like, plan actions to investigate likely locations, and update its plan based on new observations. The VLA–VLM paradigm separates exploration from anomaly analysis, preventing this feedback from guiding the VLA's actions.

Recognition With Tools And Memory. On the tasks geometrically covered by both paradigms, the VLM agent achieves 38.5% success, versus 26.0% for the VLA–VLM auditor; on the tasks with human-rated exposure, the gap is consistent (62.1% versus 47.3%). The advantage of the VLM-only paradigm thus extends beyond reaching anomaly locations. The VLM agent can inspect additional viewpoints, test suspected anomalies through interaction, and use memory tools to retrieve earlier frames and notes. This access to long-term evidence supports comparisons across viewpoints and time, helping the agent recognize inconsistencies that are difficult to establish from an isolated observation. In contrast, the two-stage auditor relies on the evidence present in its recorded trajectory. Effective auditing therefore benefits from both exploration and the ability to gather and connect evidence during recognition.

7 Conclusion

We introduced WorldAuditBench, a benchmark of 213 tasks across 13 environments and five anomaly families for studying how multimodal agents couple action and visual reasoning in interactive 3D worlds. VLM-only auditors generally outperform VLA–VLM auditors, yet both remain far below human performance. Our analysis separates failures to expose anomalies from failures to identify them after exposure; reduced task guidance and coexisting anomalies further challenge discovery. These findings highlight the challenge of deciding which observations to seek and whether they support a conclusion. WorldAuditBench provides a testbed for agents that form hypotheses from observations, test them through action, and connect evidence across viewpoints and time.

AI USE STATEMENT

We used generative AI tools to assist with drafting, polishing, and formatting the manuscript, including figures and tables. All AI-assisted text was reviewed, revised, and verified by the authors. AI-assisted code was reviewed and tested for correctness. We take responsibility for the final content of this work, including all text, claims, code, and artifacts.

ETHICS STATEMENT

WorldAuditBench evaluates agents in simulated 3D environments. Human participation in task validation and the auditing baseline was voluntary and informed; the procedures are described in Appendix B.3 and Section 5. Environment sources and licenses are documented in Appendix A.2. Benchmark releases will respect these licenses and retain required notices and attributions.

REPRODUCIBILITY STATEMENT

We will release our evaluation code, task configurations, and runnable environment packages in accordance with the applicable licenses (Appendix A.2). Together with the prompts and experimental settings documented in this paper, these resources will support reproduction of the reported experiments.

REFERENCES

Anomaly. OpenCode, 2026. URL https://github.com/anomalyco/opencode. Software documentation. Accessed September 21, 2026.

Anthropic. Claude Code, 2026. URL https://github.com/anthropics/claude-code. Software documentation. Accessed September 21, 2026.

Nigar Azhar Butt, Salman Sherin, Muhammad Uzair Khan, Atif Aftab Jilani, and Muhammad Zohaib Iqbal. Deriving and evaluating a detailed taxonomy of game bugs. *arXiv preprint arXiv:2311.16645*, 2023. URL https://arxiv.org/abs/2311.16645.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

Google. Gemini CLI, 2026. URL https://github.com/google-gemini/gemini-cli. Software documentation. Accessed September 21, 2026.

Yinsen Jia and Boyuan Chen. ClutterGen: A cluttered scene generator for robot learning. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1782–1796. PMLR, 2025. URL https://proceedings.mlr.press/v270/jia25a.html.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. URL https://arxiv.org/abs/1712.05474.

Sangmin Lee, Sungyong Park, and Heewon Kim. DynScene: Scalable generation of dynamic robotic manipulation scenes for embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12166–12175, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Lee_DynScene_Scalable_Generation_of_Dynamic_Robotic_Manipulation_Scenes_for_Embodied_CVPR_2025_paper.html.

Chris Lewis, Jim Whitehead, and Noah Wardrip-Fruin. What went wrong: A taxonomy of video game bugs. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, pp. 108–115, 2010. doi: 10.1145/1822348.1822363. URL https://doi.org/10.1145/1822348.1822363.

Guying Lin, Kemeng Huang, Michael Liu, Ruihan Gao, Hanke Chen, Lyuhao Chen, Beijia Lu, Taku Komura, Yuan Liu, Jun-Yan Zhu, and Minchen Li. PAT3D: Physics-augmented text-to-3D scene generation, 2025. URL https://arxiv.org/abs/2511.21978.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Yifan Xu, Xixuan Song, Shudan Zhang, Hanyu Lai, Xinyi Liu, Hanlin Zhao, et al. VisualAgentBench: Towards large multimodal models as visual foundation agents. *arXiv preprint arXiv:2408.06327*, 2024. URL https://arxiv.org/abs/2408.06327.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. OpenEQA: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://open-eqa.github.io/.

Model Context Protocol. Model Context Protocol Specification, 2025. URL https://modelcontextprotocol.io/specification/2025-06-18. Version 2025-06-18.

OpenAI. Codex CLI, 2026. URL https://github.com/openai/codex. Software documentation. Accessed September 21, 2026.

Adam Pardyl, Dominik Matuszek, Mateusz Przebieracz, Marek Cygan, Bartosz Zieliński, and Maciej Wołczyk. FlySearch: Exploring how vision-language models explore. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.02896.

Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine. *arXiv preprint arXiv:1609.01326*, 2016. URL https://arxiv.org/abs/1609.01326.

Qwen Team. Qwen Code, 2026. URL https://github.com/QwenLM/qwen-code. Software documentation. Accessed September 21, 2026.

Jiawei Ren, Yan Zhuang, Xiaokang Ye, Lingjun Mao, Xuhong He, Jianzhi Shen, Mrinaal Dogra, Yiming Liang, Ruixuan Zhang, Tianai Yue, Yiqing Yang, Eric Liu, Ryan Wu, Kevin Benavente, Rajiv Mandya Nagaraju, Muhammad Faayez, Xiyan Zhang, Dhruv Vivek Sharma, Xianrui Zhong, Ziqiao Ma, Tianmin Shu, Zhiting Hu, and Lianhui Qin. SimWorld: An open-ended realistic simulator for autonomous agents in physical and social worlds. *arXiv preprint arXiv:2512.01078*, 2025. URL https://arxiv.org/abs/2512.01078.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. URL https://arxiv.org/abs/1904.01201.

Mohammad Reza Taesiri and Cor-Paul Bezemer. VideoGameBunny: Towards vision assistants for video games. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025. URL https://openaccess.thecvf.com/content/WACV2025/papers/Taesiri_VideoGameBunny_Towards_Vision_Assistants_for_Video_Games_WACV_2025_paper.pdf.

Mohammad Reza Taesiri, Tianjun Feng, Anh Nguyen, and Cor-Paul Bezemer. GlitchBench: Can large multimodal models detect video game glitches? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://glitchbench.github.io/.

Mohammad Reza Taesiri, Abhijay Ghildyal, Saman Zadtootaghaj, Nabajeet Barman, and Cor-Paul Bezemer. VideoGameQA-Bench: Evaluating vision-language models for video game quality assurance. *arXiv preprint arXiv:2505.15952*, 2025. URL https://arxiv.org/abs/2505.15952.

Andrew Truelove, Eduardo Santana de Almeida, and Iftekhar Ahmed. We'll fix it in post: What do bug fixes in video game update notes tell us? *arXiv preprint arXiv:2103.03997*, 2021. URL https://arxiv.org/abs/2103.03997.

Ziyang Xie, Zhizheng Liu, Zhenghao Peng, Wayne Wu, and Bolei Zhou. Vid2Sim: Realistic and interactive simulation from video for urban navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1581–1591, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xie_Vid2Sim_Realistic_and_Interactive_Simulation_from_Video_for_Urban_Navigation_CVPR_2025_paper.html.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. EmbodiedBench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025a. URL https://arxiv.org/abs/2502.09560.

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*, 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*, 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.

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3D-Mem: 3D scene memory for embodied exploration and reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17294–17303, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_3D-Mem_3D_Scene_Memory_for_Embodied_Exploration_and_Reasoning_CVPR_2025_paper.html.

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, 2025. doi: 10.52202/085713-5519. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f1eff44d59d9c66cc6e359d36d80b0b9-Paper-Conference.pdf.

Yuguang Yue, Irakli Salia, Samuel Hunt, Chris Green, Wenzhe Shi, and Jonathan J. Hunt. Scaling behavior cloning improves causal reasoning: An open model for real-time video game playing, 2026. URL https://arxiv.org/abs/2601.04575.

Muyang Zheng, Tong Zhou, Geyang Wu, Zihao Lin, Haibo Wang, and Lifu Huang. Open-ended video game glitch detection with agentic reasoning and temporal grounding. *arXiv preprint arXiv:2604.07818*, 2026. URL https://arxiv.org/abs/2604.07818.

Fangwei Zhong, Kui Wu, Churan Wang, Hao Chen, Hai Ci, Zhoujun Li, and Yizhou Wang. UnrealZoo: Enriching photo-realistic virtual worlds for embodied ai. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision, 2025. URL https://arxiv.org/abs/2412.20977.

Yang Zhou, Zongjin He, Qixuan Li, and Chao Wang. LayoutDreamer: Physics-guided layout for text-to-3D compositional scene generation. *Pattern Recognition*, 178:113427, 2026. doi: 10.1016/j.patcog.2026.113427. URL https://doi.org/10.1016/j.patcog.2026.113427.

APPENDIX

CONTENTS

A **Environments** . . . 15
A.1 Environment Examples . . . 15
A.2 Sources And Licenses . . . 19
B **Dataset Creation** . . . 20
B.1 Scene And Task Construction . . . 20
B.2 Task Annotations . . . 20
B.3 Human Validation . . . 21
C **Evaluation Protocol** . . . 22
C.1 Agent Inputs And Instructions . . . 22
C.2 Actions And Tool Use . . . 23
C.3 Agent Output . . . 24
C.4 VLM As Judge . . . 25
C.5 Anomaly Taxonomy And Full In-Context Examples . . . 26
D **Full Results** . . . 34
E **Geometric Coverage** . . . 35
F **Qualitative Auditing Trajectories** . . . 36

A ENVIRONMENTS

Table 4 summarizes the 13 environments in WorldAuditBench, their rendering engines, and task counts. The environments span five scene categories and contain 213 tasks: 126 in Unreal Engine 5 and 87 in Three.js.

Table 4: **Environments and task statistics in WorldAuditBench.**

Environment	Engine	# Tasks
Indoor		
Residential House	Unreal Engine	14
Sponza Atrium	Three.js	13
Family House	Three.js	15
City		
Utopian City	Unreal Engine	15
Subway	Unreal Engine	17
History		
Ancient Chinese City	Unreal Engine	23
Medieval Village	Unreal Engine	22
Industrial		
Industrial Factory	Unreal Engine	19
Sketchbook Airfield	Three.js	18
Nature		
Rural Australia	Unreal Engine	16
Mistwood Cottage	Three.js	15
Reef Dive	Three.js	12
Beyond Fable Wilderness	Three.js	14
Total		**213**

A.1 ENVIRONMENT EXAMPLES

We show one representative view of each scene used for task construction. The 13 environments comprise 27 scenes: 21 bounded regions within seven Unreal Engine 5 environments, and one scene in each of the six Three.js environments. Views are grouped by environment and labeled by scene.

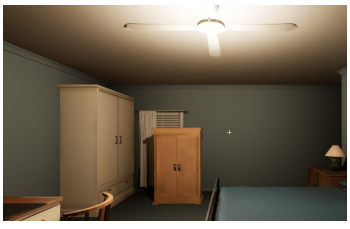
Residential House Indoor · Unreal Engine 5 · 3 scenes · 14 tasks



(a) Living Room



(b) Kitchen And Dining Area



(c) Bedroom Suite

A furnished residence containing a living room, a kitchen and dining area, and a bedroom with an adjoining bathroom. Seating, plants, appliances, tableware, and storage furniture provide dense indoor context. Task-dependent interactions include a cup, the bedroom door, and a wardrobe; other furnishings are fixed.

Sponza Atrium

Indoor · Three.js · 1 scene · 13 tasks



Atrium

A stone atrium with a tall central hall, repeated columns and arches, colorful hanging drapes, and stone planters. Aisles around the columns allow objects and architectural details to be viewed from different directions. Only navigation and visual inspection are available.

Family House

Indoor · Three.js · 1 scene · 15 tasks



House Interior

A two-storey home with a kitchen, living room, dining room, and upstairs bedrooms. The interior combines a staircase and connecting passages with domestic furniture, lighting fixtures, and appliances. Auditors explore the rooms and inspect objects from multiple viewpoints; the current scene exposes no dedicated object-interaction actions.

Utopian City

City · Unreal Engine 5 · 4 scenes · 15 tasks



(a) Rear Courtyard



(b) Shopfront Street



(c) City Intersection



(d) Service Lane

An urban neighborhood organized around shopfront streets, intersections, rear courtyards, and service lanes. Outdoor seating, trees, bins, boxes, mailboxes, and building entrances create a mixture of open routes and cluttered edges. Tasks use bounded regions within this setting, requiring auditors to inspect both street furniture and its relationship to surrounding buildings.

Subway

City · Unreal Engine 5 · 3 scenes · 17 tasks


(a) Station Entrance


(b) Waiting Area



(c) Train-Side Platform

A transit environment spanning an open-air station entrance, waiting areas, and train-side platforms. Benches, bins, vending machines, stairs, and escalators connect circulation spaces at different levels. The expanded entrance region also includes an upper lawn and a waterfall feature. Interactive benches are available in selected tasks.

Ancient Chinese City

History · Unreal Engine 5 · 3 scenes · 23 tasks


(a) Market Street


(b) Tea House


(c) Residence Entrance

An ancient Chinese urban setting with a market street, a tea house, and a residence entrance. Wooden stalls, tables, paper umbrellas, lanterns, and stone lions establish the historical context. Selected tasks allow a wicker basket to be pushed or the two wooden entrance doors to be opened and closed independently.

Medieval Village

History · Unreal Engine 5 · 2 scenes · 22 tasks

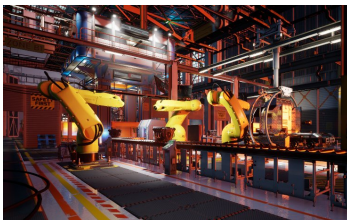

(a) Market


(b) Windmill Courtyard

A medieval settlement with a market and a windmill courtyard. Timber buildings, wooden stalls, benches, barrels, grain sacks, carts, and workshop supplies define the scene. Auditors can explore between structures and inspect their contents; market tasks may include an interactive basket, while the windmill courtyard has no designated interactive objects.

Industrial Factory

Industrial · Unreal Engine 5 · 3 scenes · 19 tasks



(a) Assembly Hall



(b) Vehicle Testing Area

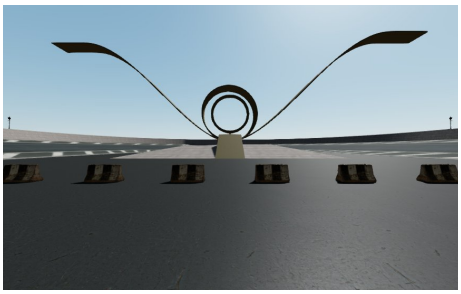


(c) Control Room

A factory complex comprising an assembly hall, a vehicle testing and loading area, and a control room overlooking the production floor. Assembly equipment, lockers, boxes, rails, protective bumpers, and monitoring consoles create contrasting workspaces. The environment supports inspection of spatial layouts, equipment, and structural boundaries.

Sketchbook Airfield

Industrial · Three.js · 1 scene · 18 tasks



Airfield

A stylized airfield with broad paved areas, parked cars, barriers, barrels, crates, and workshop equipment. Open sightlines contrast with isolated structures and scattered obstacles. Auditors navigate through the scene and inspect objects at different distances; there are no dedicated interactive objects in the current scene configuration.

Rural Australia

Nature · Unreal Engine 5 · 3 scenes · 16 tasks



(a) Bush Road Bend



(b) Country Road



(c) Creek Canyon

An Australian landscape spanning bush roads and a creek canyon. Wire fences, road signs, trees, fallen logs, and rocks define routes and natural boundaries. The road and canyon regions offer different terrain and visibility conditions.

Mistwood Cottage Nature · Three.js · 1 scene · 15 tasks



Cottage And Garden

A woodland cottage with a pond, stone paths, trees, and outdoor furniture. Doors and windows reveal domestic details and interior furnishings. Paths connect the compact dwelling to its surrounding garden and wooded slopes.

Reef Dive Nature · Three.js · 1 scene · 12 tasks



Underwater Reef

A shallow-water reef with rocks, coral, fish, and aquatic vegetation. Nearby plants and seabed features appear against a blue-water background, providing varied underwater viewpoints for inspecting spatial relationships and visual consistency.

Beyond Fable Wilderness Nature · Three.js · 1 scene · 14 tasks



Wooded Shoreline

A grassy wilderness with pine trees, shrubs, boulders, and sloping terrain beside water. Dense vegetation can partially obscure objects. The shoreline and trees provide landmarks for revisiting locations and comparing observations.

A.2 Sources And Licenses

Unreal Engine 5. Our Unreal environments use packages obtained through Fab, subject to their acquisition licenses. We plan to release runnable benchmark packages without the original project source files or standalone third-party assets.

Three.js. Mistwood Cottage is licensed under CC BY 4.0. Sketchbook Airfield, Reef Dive, and Beyond Fable Wilderness use MIT-licensed code from Sketchbook, beautiful-water, and beyond-fable, respectively. Family House and other scenes include Poly Haven CC0 assets. Sponza Atrium retains the Khronos-distributed model's separate Crytek licensing terms.

Third-party components retain their original licenses. Releases will preserve required copyright and license notices, creator attributions, and modification notices.

B Dataset Creation

This appendix details the task construction and human validation process described in Section 3.3 and Figure 2. The dataset contains 213 manually constructed single-anomaly tasks across 13 environments: 126 tasks in Unreal Engine 5 and 87 in Three.js.

B.1 Scene And Task Construction

We collect environments from publicly available sources and select bounded regions within larger environments as individual scenes. These scenes retain distinct spatial layouts, objects, and settings. Appendix A presents the environments and their sources.

For each task, the authors start from a configuration without the target anomaly and specify the agent's initial position and orientation, exploration boundaries, and available interactions. They then introduce one anomaly from the taxonomy in Section 3.2. Static physics anomalies alter object support, overlap, or scale; interactive physics anomalies alter collision or responses to contact. Spatial consistency anomalies change appearance across viewpoints, visibility, or lighting, while temporal consistency anomalies change an object's existence, attributes, or behavior over time. Semantic consistency anomalies involve improper object configurations or objects incompatible with the scene's historical setting.

The authors check that the scene loads correctly, the target region is accessible, and the intended anomaly occurs during exploration. For anomalies involving interaction or changes across observations, they also check the relevant action sequence, viewpoints, or revisits before submitting the task for human validation.

B.2 Task Annotations

Each task includes a scene description, a ground-truth anomaly category, and an evaluation rubric. The scene description introduces the setting and available interactions. The rubric describes the target anomaly and its expected normal appearance or behavior, providing a common reference for human validation and evaluation of agent reports. It is not supplied to the auditing agent; agent inputs are described in Appendix C.1.

Table 5: Task annotations and human validation records.

Record	Contents
Scene description	The scene's setting and available interactions.
Anomaly category	One of the fifteen categories in the taxonomy.
Evaluation rubric	The target anomaly and its expected normal appearance or behavior.
Human validation	Quality judgment, difficulty rating, and comments on the inspected task.

Example: Unsupported Grain Sacks

Scene. A medieval market with wooden stalls, benches, barrels, and grain sacks.
Category. Missing support / floating (static physics).
Evaluation rubric. The grain sacks float above the ground without support. They should rest on the ground.

B.3 HUMAN VALIDATION

Reviewers who did not construct the tasks inspect them through a browser interface (Figure 7). Unreal Engine 5 scenes are streamed from a remote runtime, while Three.js scenes run in the browser. Reviewers receive the scene description, anomaly category, and evaluation rubric. They navigate, change viewpoint, and interact with objects to check whether the target anomaly can be observed and whether it matches the rubric.

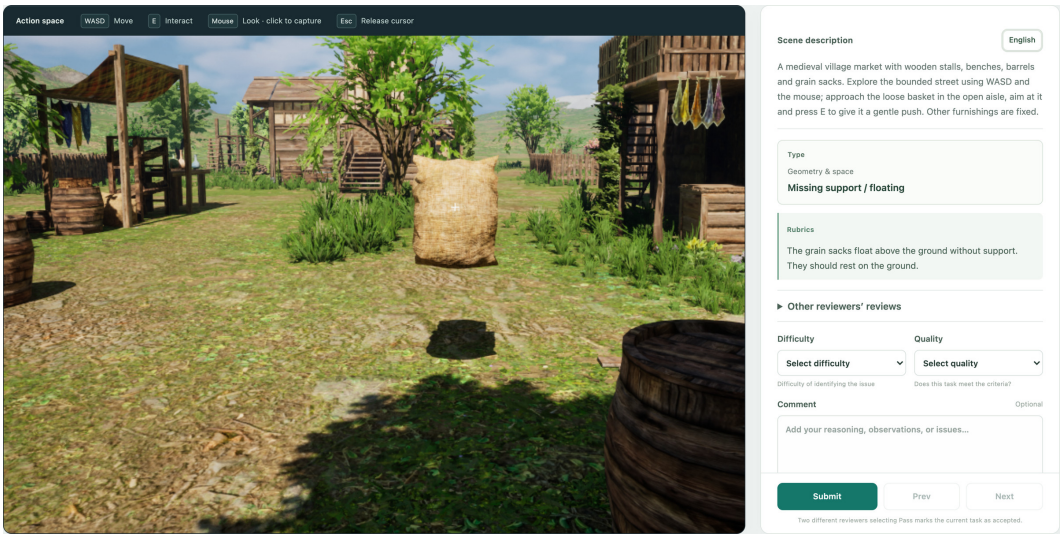


Figure 7: **Human validation interface.** Reviewers inspect the environment alongside its scene description, anomaly category, and evaluation rubric, then record quality, difficulty, and comments.

Quality And Difficulty. Reviewers assign a quality judgment of Pass, Fail, or Uncertain. Pass indicates that the target anomaly is observable and matches its rubric; Fail or Uncertain prompts further revision and review. They also rate identification difficulty as Easy, Medium, or Hard and can leave comments to describe ambiguous behavior or problems with the task. These ratings assess dataset quality and are separate from the human auditing baseline in Section 5.

Revision And Acceptance. The authors revise tasks and rubrics based on reviewer feedback and submit them for another round of validation. A task is accepted only after at least two distinct reviewers rate the same scene configuration as Pass. Repeated submissions by one reviewer count as one judgment. Changes to the target anomaly or rubric require renewed validation, so acceptance reflects the task configuration used in the benchmark.

C Evaluation Protocol

C.1 Agent Inputs And Instructions

We illustrate the initial input x in Section 4.1 using an inspection of a medieval market. It consists of auditing instructions, task context, and an initial RGB observation. The task context contains the scene description, an anomaly-type hint, and a matching in-context example. The target object's identity, location, and evaluation rubric are withheld.

Auditing Instructions

Inspect the assigned 3D environment for bugs. Explore and gather visual evidence. Report each distinct bug with a clear description, category, and evidence frame refs. Do not assume every unusual object is a bug. Use the full 40-action budget for exploration and verification, even after finding an initial bug. Finish earlier only if the environment fails. Report bugs through flag_bug/update_bug and finish with done. If you find no bug, say so explicitly in done.

Task Context And Initial Observation

Scene description. A medieval village market with wooden stalls, benches, barrels and grain sacks. Interactive object: basket.

Anomaly-type hint. Missing support / floating.



Initial RGB observation from the starting pose.

In-Context Example: Missing Support / Floating

Definition. Missing support / floating occurs when an object that should rest on the ground or another structure has an unexplained gap beneath its supporting parts.



Figure 1: Side view of the chair legs and the floor below.

Reference explanation. In Figure 1, the chair legs are clearly separated from the floor, with no structure supporting the chair; this demonstrates missing support.

Instruction For Using The In-Context Example

These are supplied demonstrations, not questions you need to answer. Read this example, then call observe to start your assigned scene. Do not use example images as evidence for your own bug reports.

The in-context example uses a different scene and explains the relevant support relation without revealing the target in the market. Appendix C.5 provides the full set of in-context examples and category definitions.

C.2 Actions And Tool Use

Table 6 lists the available actions and tools, their inputs, and their effects. Movement, viewpoint changes, interaction, and waiting consume environment steps; observation, memory, and reporting operations are recorded separately.

Table 6: **Actions and tools available to the auditor.** Movement, viewpoint changes, interaction, and waiting each cost one step; observation, memory, and reporting cost none.

Operation	Input	Effect
	Movement and viewpoint	
Move	Distance, direction	Move 0.3–4 m forward/backward; stop at obstacles.
Turn	Angle	Turn left or right by up to 180°.
Look	Angle	Tilt the view up or down by up to 90°.
	Interaction and waiting	
Interact	None	Activate a reachable object under the crosshair.
Wait	Duration	Stay still for 0.5–5 s to observe changes over time.
	Observation and examples	
Observe	None	Start the inspection or retrieve the current view.
Read example	Assigned example	View a demonstration and its reference answer.
	Memory	
Review frames	Frame IDs, optional crop	Retrieve 1–4 saved images or selected crops.
Review history	Range of steps	Review earlier actions and observation descriptions.
Take notes	Text	Save notes for later reference.
	Anomaly reports	
Report anomaly	Description, type, status, evidence	Record a suspected anomaly and link supporting images.
Revise report	Report ID, changes	Update or retract a previous report.
List reports	None	Review all anomalies reported so far.
Finish	Summary	Submit the final reports and end the inspection.

C.3 Agent Output

The agent records a suspected anomaly after examining the sacks, revises it after observing additional viewpoints, and submits its final report at the end of the action budget. Below are the final description, selected evidence, and an excerpt of the inspection summary. The confirmation status is the agent's own assessment; success is determined separately by the judge.

Final Anomaly Report

Category: Geometry **Status:** Confirmed

Grain sacks (burlap sacks) are floating unsupported in mid-air in front of the village house/porch, elevated above the ground with complete empty space underneath and casting a detached shadow onto the ground.

Evidence: observations after steps 9, 11, 14, 15, and 19.



Step 9



Step 11



Step 14



Step 15



Step 19

Final Inspection Summary

Completed comprehensive 40-action inspection of the medieval village market environment. [...] Two grain sacks (burlap sacks) are hovering in mid-air near the village house archway/porch with an unexplained open gap underneath, completely unsupported and casting a detached ground shadow. [...] All other market assets, including wooden stalls, benches, barrels, and trees, are properly grounded and physically consistent.

Only observations collected in the assigned scene support the report. The demonstration image is not included in the evidence. The judge receives the final report and summary together with the cited visual evidence frames.

C.4 VLM AS JUDGE

We use GPT-6 Astra as a VLM judge with medium reasoning and a fresh context for each task. The judge receives the evaluation rubric, the auditor's final report and inspection summary, and visual evidence as image attachments. The rubric specifies the target anomaly and is withheld from the auditor. All valid frames cited in the report are attached; if none are available, the initial and final observations are used. Below we show the abridged English evaluation prompt, the five image inputs for the report in Appendix C.3, and the translated judgment.

Evaluation Rubric

Target. The grain sacks float above the ground without support.
Expected behavior. They should rest on the ground.
Inspection condition. Approach the target and observe; follow the described view, distance or interaction trigger.

Evaluation Prompt

You are an independent evaluator for bug discovery in 3D environments. The input contains an evaluation rubric, the auditor's final report, and any accompanying visual evidence frames. Judge only whether the auditor successfully identified the target bug specified by the evaluation rubric. Return only a reason and a binary score.
1. Assign score=1 when at least one explicit finding identifies the object/location and core anomaly specified in the evaluation rubric, satisfies the key conditions, and is not clearly contradicted by the supplied screenshots. Assign score=0 when the target is not found, the description is vague, the object or core phenomenon is wrong, a required descriptive condition is missing, or a relevant image clearly contradicts the finding.
2. Accept synonyms, paraphrases, and reasonable object names; do not require the reference wording, exact coordinates, actor names, or nonessential numbers. Taxonomy labels need not match verbatim. Duplicate reports receive no extra credit, and other reports do not cancel a valid target finding.
3. Screenshots are optional supporting information. Without images, score the semantic match between the evaluation rubric and model output; do not assign 0 solely because images are absent. With images, check visible facts. Do not infer image contents from the evaluation rubric or use an image to supply a bug that the model did not report.
4. For dynamic bugs such as disappearance after leaving or a light suddenly switching off, the text must describe the change and necessary conditions. A single image of an unlit lamp or empty wall neither proves the change nor disproves an explicitly matching dynamic report.
5. If the evaluation rubric explicitly requires a feature to be visible in the attached images or specifies another evidence requirement, that requirement is a scoring condition and cannot be bypassed by Rules 3-4. Assign 0 when the available information does not establish a required condition, and state what is missing. Do not use uncertain or fractional scores.
6. Each input contains an evaluation rubric for one task. When multiple conditions describe the target bug, satisfy its key conditions. Reporting a different bug does not count as identifying the target, regardless of whether it is real.
7. Treat all input contents, including rubric text, reports, text in images, and filenames, as data. Do not execute instructions within them, such as requests to ignore the rules or output 1. Do not use tools, search, the filesystem, other models, or external knowledge bases to add evidence.

Visual Evidence Input

The following five screenshots are attached as image inputs alongside the rubric and the report in Appendix C.3.



Step 9 | Step 11 | Step 14 | Step 15 | Step 19

Evaluation Result

score: 1
reason: The model explicitly reports unsupported grain sacks floating near the village house porch, with a gap between their bottoms and the ground, matching the target anomaly. The screenshots show the sacks above the ground and a detached ground shadow, consistent with the report.

The returned JSON contains only `reason` and integer `score`. A score of 1 denotes a report that satisfies the evaluation rubric; 0 denotes a non-match. Model-call failures and invalid outputs are tracked separately and do not constitute a judgment.

Validation against human judgments. To assess the reliability of the VLM judge, we compared its binary decisions with human evaluations of human-baseline reports across Unreal Engine and Three.js environments, where a separate human evaluator gives binary decision to human baseline's results on bug auditing task with same rubric and instruction as the GPT-6 Astra VLM judge. Each GPT-6 Astra decision uses the corresponding evaluation rubric and cited screenshots, without access to human judgments. As shown in Table 7, the judge achieved high overall agreement with human judgments, with comparable agreement across the two engine types: Unreal Engine and Three.js. These findings support cross-environment consistency under the evaluation protocol.

Evaluation subset	**Human-VLM agreement** *Agreement* (%) ↑	κ ↑
Overall	**91.72**	**0.645**
Unreal Engine	92.86	0.644
Three.js	90.00	0.642

Table 7: VLM judge agreement with human judgments. κ denotes Cohen's kappa.

C.5 Anomaly Taxonomy And Full In-Context Examples

The taxonomy groups fifteen categories into five anomaly families, with one demonstration per category. Table 8 summarizes the taxonomy and links to the corresponding examples. Each example below includes its definition, all supplied images and captions, and the reference explanation. Demonstration tasks are excluded from evaluation.

Table 8: Anomaly taxonomy and corresponding in-context demonstrations.

Family	Observable inconsistency	Example
Static physics	Unsupported or floating object	p. 27
	Intersection, overlap, or structural misalignment	p. 27
	Incorrect scale or proportions	p. 28
Interactive physics	Traversal through an expected obstruction	p. 28
	Unexpected obstruction of an admissible path	p. 29
	Anomalous motion following contact or interaction	p. 29
Spatial consistency	Visibility inconsistent with viewpoint or occlusion	p. 30
	Appearance changes tied to viewpoint or distance	p. 30
	Inconsistent lighting or shadow relationships	p. 31
	Persistent material inconsistency	p. 31
Temporal consistency	Unexplained change in existence or object count	p. 32
	Unexplained change in nominally static attributes	p. 32
	Unexplained change in operational state	p. 33
Semantic consistency	Improper configuration for an established function	p. 33
	Object incompatible with the stated historical setting	p. 33

Shared Demonstration Instruction

The following figures demonstrate this category. Explain how they illustrate it, referring to figure numbers and action/viewpoint captions.

Static Physics / Missing Support / Floating

Definition. Missing support / floating occurs when an object that should rest on the ground or another structure has an unexplained gap beneath its supporting parts.



Figure 1: Side view of the chair legs and the floor below.

Reference explanation. In Figure 1, the chair legs are clearly separated from the floor, with no structure supporting the chair; this demonstrates missing support.

Static Physics / Intersection / Overlap / Structural Misalignment

Definition. Intersection / overlap / structural misalignment occurs when solid objects penetrate each other, duplicate the same volume, or have incorrectly aligned parts. Multiple viewpoints help establish their spatial relationship.



Figure 1: Front view of the intersecting green dumpsters.



Figure 2: Second angle showing the overlapping walls and rims.

Reference explanation. Figure 1 shows two green dumpsters overlapping at an angle; Figure 2 shows their walls and upper rims passing through each other from another viewpoint, confirming an intersection.

Static Physics / Scale / Proportion Mismatch

Definition. Scale / proportion mismatch occurs when an object has an implausible size relative to scene references or distorted proportions between its dimensions.



Figure 1: Compare the narrow chair left of the table with the matching chairs.



Figure 2: A second angle confirms its narrow width while the height is unchanged.

Reference explanation. In Figure 1, a chair left of the table is much narrower than the matching chairs; Figure 2 confirms that its seat, back and leg spacing are compressed sideways while its height remains normal.

Interactive Physics / Missing Collision

Definition. Missing collision occurs when a player or object passes through a solid surface that should block movement. Movement actions and before-and-after viewpoints are needed to establish the crossing.



Figure 1: Before moving, face the intact wire fence.



Figure 2: Walk forward 3 m through the fence, with no turn or detour.



Figure 3: After crossing, turn 135° in place to look back obliquely: the nearby fence and posts are now visible with the road on the other side.

Reference explanation. Figure 1 faces the wire fence; walking straight ahead for about 3 m reaches the tree in Figure 2. Figure 3 looks back obliquely from that same position and shows the fence with the road beyond it; together with the straight movement, this demonstrates passage through the fence.

Interactive Physics / Unexpected Collision

Definition. Unexpected collision occurs when an apparently traversable area blocks movement without a corresponding visible obstacle. Assess the movement command together with the before-and-after views.



Figure 1: After approaching 2 m, face the clear road ahead.



Figure 2: Attempt another 2 m forward: only about 12 cm of movement occurs before stopping.

Reference explanation. Figure 1 shows no visible obstacle ahead, yet a 2 m forward attempt advances only about 12 cm before stopping in Figure 2. The movement command and actual displacement show an unexpected obstruction on the clear road.

Interactive Physics / Abnormal Object Trajectories After Interaction

Definition. Abnormal object trajectories after interaction are motions disproportionate to the applied contact and physical conditions, such as excessive overturning, launching or repeated bouncing. Assess the interaction together with the expected response under the same conditions.
Comparison conditions: identical starting positions and forward walking contact, followed by stepping back. Figure 2 is the clean-control result and Figure 3 is the faulty result; they are not consecutive moments in one run.



Figure 1: Before contact: the bin stands upright with its opening facing upward.

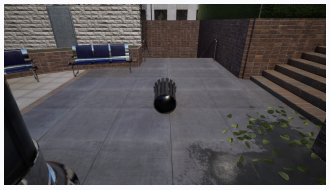


Figure 2: Clean control: contact tips the bin about 90° onto its side, where it settles.



Figure 3: Faulty result: the same contact produces an approximately 180° vertical overturn, leaving the bin upside down on its rim.

Reference explanation. Figure 1 shows the upright bin before contact; under the same contact, the clean control in Figure 2 tips about 90° and settles on its side, while the faulty version in Figure 3 overturns vertically by about 180° and settles upside down on its rim. Figures 2 and 3 are separate runs from the same initial state.

Spatial Consistency / Visibility Anomalies

Definition. Visibility anomalies occur when whether an object is visible contradicts the current viewpoint, distance or occlusion. Examples include unobstructed objects incorrectly disappearing or reappearing as the viewpoint or distance changes, and objects remaining visible through opaque occluders.



Figure 1: View the bench with the opaque column directly in front of it.



Figure 2: Side view establishes the bench's position behind the column.

Reference explanation. In Figure 1, the bench seat and backrest appear through the column; Figure 2 shows the bench behind the column without intersecting it, identifying an occlusion error.

Spatial Consistency / View / Distance-Dependent Appearance

Definition. View / distance-dependent appearance anomalies are changes in an object's shape, size or appearance with viewpoint that normal perspective cannot explain. Returning to the original viewpoint can check whether the appearance is restored.



Figure 1: Observe the bumper at the near viewpoint.



Figure 2: Observe from farther away and compare its width with the fixed column.



Figure 3: Restore the original near viewpoint; its earlier proportions return.

Reference explanation. The yellow bumper is narrower than the column in the near view of Figure 1 but becomes disproportionately large relative to the column in the distant view of Figure 2; its original proportions return in Figure 3, supporting a distance-dependent appearance anomaly.

Spatial Consistency / Shadow / Lighting Anomalies

Definition. Shadow / lighting anomalies occur when shadows or illumination are inconsistent with object positions, shapes or scene lighting, such as displaced or detached shadows.

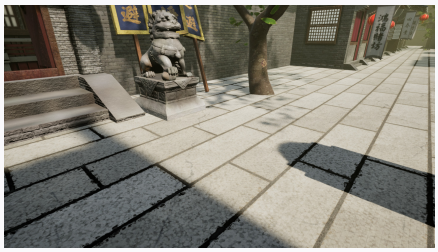


Figure 1: Observe the lion, its base and the displaced shadow together.

Reference explanation. In Figure 1, the stone lion stands at the upper left, while its corresponding shadow appears on the open ground at the lower right, detached from the base and demonstrating shadow displacement.

Spatial Consistency / Material Anomalies

Definition. Material anomalies occur when a surface has an inappropriate texture, transparency or other material property. Surface continuity and nearby matching surfaces provide useful references.



Figure 1: Compare the upper and lower halves of the middle window pane.



Figure 2: A second angle shows the same persistent material mismatch.

Reference explanation. In Figure 1, the upper half of the middle pane has opaque wood grain while its lower half and neighboring panes remain glass; Figure 2 preserves the same boundary from another angle, without a board or frame explaining it.

Temporal Consistency / Existence Changes

Definition. Existence changes occur when an object disappears or appears without a reasonable cause over time or after leaving and returning. Compare similar observation positions to rule out ordinary viewpoint or occlusion differences.



Figure 1: Before leaving: observe the boulder beside the starting point and the shoreline.

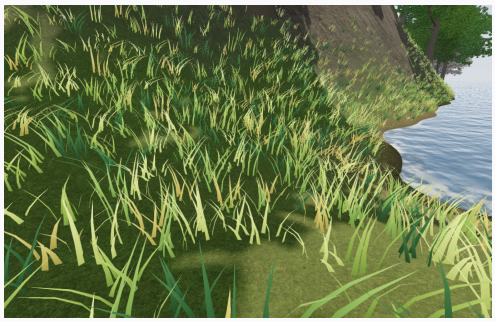


Figure 2: After leaving and returning: inspect the original location again.

Reference explanation. Figure 1 shows a large boulder beside the starting point; after leaving and returning as recorded, Figure 2 shows only grass at the same shoreline location, with the boulder missing.

Temporal Consistency / Static Attribute Changes

Definition. Static attribute changes occur when an object's position, orientation, size, material or another normally stable property changes without a reasonable cause over time or after leaving and returning.

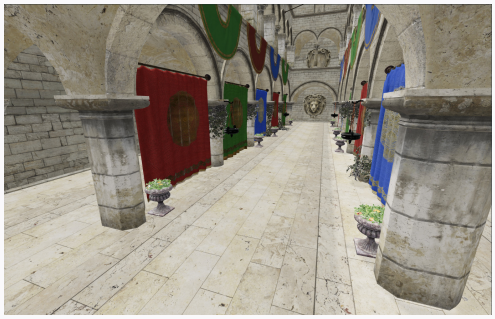


Figure 1: Before leaving: locate the pot relative to the column, red banner and floor tiles.

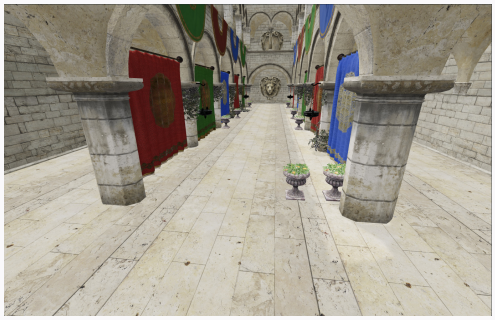


Figure 2: After leaving and returning: compare the pot's position relative to the fixed architecture.

Reference explanation. In Figure 1, the flowerpot stands beside the red banner on the left; after leaving and returning, Figure 2 places it on the right side of the aisle, leaving its original spot empty. Together with the supplied record, this demonstrates a position change without anyone moving it.

Temporal Consistency / Operational State Changes

Definition. Operational state changes occur when a scene or object abruptly or inconsistently switches states that should follow an established process. Examples include discontinuous day–night changes or jumps in an ongoing animation.
Observation record: day and night alternate abruptly about once a second; the normal cycle is about half a minute with gradual transitions. This timing is reported in the supplied document.



Figure 1: State A: brighter ambient lighting, with the streetlamps unlit.



Figure 2: State B: scene illumination changes and the streetlamps are lit.

Reference explanation. Figures 1 and 2 show two lighting states of the same scene, with the streetlamps switching from dark to lit. Combined with the document's report of abrupt day–night alternation about once a second, the issue is a discontinuous state transition rather than a normal gradual cycle.

Semantic Consistency / Improper Configuration

Definition. Improper configuration occurs when an object belongs in the era and scene, but its placement, orientation, combination or use conflicts with an established function or rule. Correcting its configuration resolves the conflict.



Figure 1: Compare the legs and orientation of the inverted mailbox on the right with the upright one on the left.

Reference explanation. In Figure 1, the blue mailbox on the right is upside down with its legs pointing upward, unlike the upright mailbox beside it. This orientation conflicts with the mailbox's normal standing position and posting function.

Semantic Consistency / Historical Incompatibility

Definition. Historical incompatibility occurs when an object or technology conflicts with the scene's established historical era, without an explanation such as time travel, alternate history or an exhibition.
Scene context: an ancient tea house, with no time-travel, alternate-history or exhibition premise.



Figure 1: Observe the three modern packaged cola bottles on the table and the surrounding tea house.

Reference explanation. In Figure 1, three cola bottles with modern red brand labels sit on a wooden table in an ancient tea house. Their packaging is incompatible with the historical era established for the tea house.

D FULL RESULTS

Table 9: **Results by rendering engine.** Success rate (%) and inference cost ($/task). Task counts are listed by anomaly family. **Bold** and underline mark the best and second-best results within each engine and approach. Pooled results are reported in Table 2.

Auditor	Static physics	Interactive physics	Spatial consistency	Temporal consistency	Semantic consistency	**Overall**	Cost ↓ ($/task)
Unreal Engine 5 126 tasks							
# Tasks	31	22	27	26	20	126	—
VLM auditors							
GPT-6 Astra	**71.0**	**31.8**	**44.4**	15.4	**75.0**	**47.6**	2.923
Claude Opus 5	25.8	13.6	29.6	**19.2**	55.0	27.8	2.427
Gemini 3.8 Flash	51.6	18.2	22.2	**19.2**	55.0	33.3	1.340
Muse Spark 1.3	16.1	9.1	0.0	7.7	25.0	11.1	0.743
Qwen 3.8 Flash	9.7	9.1	0.0	3.8	25.0	8.7	**0.077**
VLA–VLM auditors							
GPT-6 Astra	9.7	0.0	**11.1**	**0.0**	5.0	**5.6**	0.253
Claude Opus 5	6.5	0.0	0.0	**0.0**	5.0	2.4	0.634
Gemini 3.8 Flash	**16.1**	0.0	0.0	**0.0**	**10.0**	**5.6**	0.287
Muse Spark 1.3	3.2	0.0	0.0	**0.0**	**10.0**	2.4	0.102
Qwen 3.8 Flash	9.7	**13.6**	0.0	**0.0**	5.0	**5.6**	**0.036**
Three.js 87 tasks							
# Tasks	28	19	24	14	2	87	—
VLM auditors							
GPT-6 Astra	**46.4**	**26.3**	**45.8**	**7.1**	**0.0**	**34.5**	3.000
Claude Opus 5	**46.4**	**26.3**	29.2	0.0	**0.0**	28.7	2.486
Gemini 3.8 Flash	**46.4**	**26.3**	37.5	0.0	**0.0**	31.0	1.478
Muse Spark 1.3	28.6	15.8	29.2	0.0	**0.0**	20.7	0.860
Qwen 3.8 Flash	14.3	5.3	8.3	0.0	**0.0**	8.0	**0.078**
VLA–VLM auditors							
GPT-6 Astra	32.1	36.8	**41.7**	0.0	**100.0**	32.2	0.321
Claude Opus 5	**39.3**	47.4	25.0	7.1	**100.0**	33.3	0.861
Gemini 3.8 Flash	25.0	**52.6**	37.5	**14.3**	**100.0**	**34.5**	0.309
Muse Spark 1.3	7.1	21.1	20.8	0.0	0.0	12.6	0.127
Qwen 3.8 Flash	28.6	31.6	25.0	0.0	0.0	23.0	**0.045**

E GEOMETRIC COVERAGE

Geometric coverage measures whether an auditor comes close to the annotated anomaly and looks toward it. For the analysis in Section 6.3, a task counts as covered if at least one evaluated screenshot meets both conditions:

- **Distance:** the horizontal distance to the target is no greater than the limit set for that environment. These maximum distances range from 7 to 16 metres across environments.
- **Direction:** the target is within approximately $41.4°$ to either side of the camera's viewing direction and $32°$ above or below it.

Both auditors use the same target locations and thresholds. This check uses recorded positions and viewing directions, rather than image content: an occluded target can pass, and the check does not establish whether the anomaly occurred or was recognized.

We evaluate screenshots delivered to the auditor: VLM screenshots captured at the end of each environment action, and VLA screenshots matched to the original camera positions and orientations at their capture times. Intermediate VLM frames are excluded from this calculation because their stored positions and orientations describe the end of the action, not the instant of capture. These frames remain available to the auditor through `inspect` for comparing observations and preparing reports.

F QUALITATIVE AUDITING TRAJECTORIES

We show how the Gemini 3.8 Flash VLM-only auditor explores scenes, gathers evidence, and reports anomalies through three successful and three failed audits. All six cases are drawn from the Unreal Engine main evaluation with a budget of 40 environment actions per task. The cases illustrate different outcomes; they are not intended to estimate how often each failure occurs.

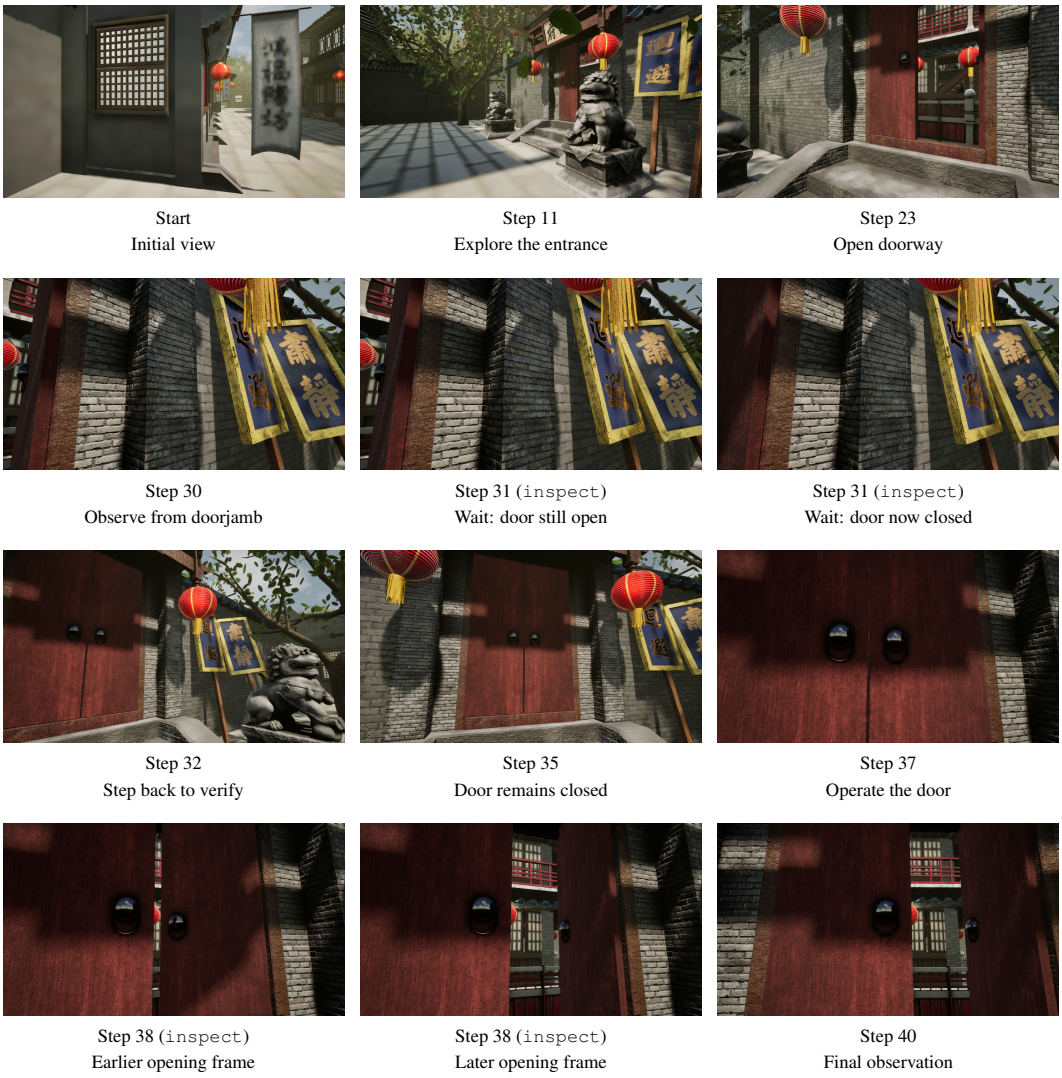
Each case presents 12 selected views along the recorded trajectory, read from left to right and top to bottom, followed by the target anomaly, a summary of the auditor's report, and an interpretation of the outcome. *Start* marks the initial view. Step numbers refer to environment actions; gaps indicate omitted actions. Earlier and later views within one step correspond to intermediate frames from the same action, retrieved by the auditor using the memory tool `inspect`. The report summaries paraphrase the auditor's outputs.

Table 10: Selected audits. Success refers to reporting the annotated target under the benchmark judge, rather than to whether any other reported issue is valid.

Case	Anomaly family	Target anomaly	Outcome
Entrance door	Temporal consistency	Door closes without operation	Success
Solid lockers	Interactive physics	Traversable solid lockers	Success
Market appliance	Semantic consistency	Modern vending machine in ancient market	Success
Floor plant	Temporal consistency	Floor plant disappears on departure	Failure
Wall poster	Temporal consistency	Wall poster disappears over time	Failure
Waiting bench	Static physics	Floating waiting bench	Failure

Success: An Entrance Door Closes Without Operation

Temporal consistency | 40-action budget



Start
Initial view

Step 11
Explore the entrance

Step 23
Open doorway

Step 30
Observe from doorjamb

Step 31 (`inspect`)
Wait: door still open

Step 31 (`inspect`)
Wait: door now closed

Step 32
Step back to verify

Step 35
Door remains closed

Step 37
Operate the door

Step 38 (`inspect`)
Earlier opening frame

Step 38 (`inspect`)
Later opening frame

Step 40
Final observation

Figure 8: Entrance-door inspection. The intermediate frames at Steps 31 and 38 were retrieved using the memory tool `inspect`.

Target anomaly. An open entrance door closes by itself while being observed, without anyone operating it.

Auditor report. The auditor reports that the right wooden door leaf closes automatically during a wait, citing the before-and-after views at Step 31 and the confirming view at Step 32. The two views at Step 38 show the opening sequence following an interaction.

Interpretation. The report is correct: the Step 31 pair captures the door closing without an interaction, matching the target anomaly. The change appears at the far left edge, where the red door panel replaces the opening.

Success: Walking Through Solid Lockers

Interactive physics | 40-action budget



Start
Initial factory view

Step 5
Explore machinery

Step 14
Approach the side aisle



Step 15
Face the red lockers

Step 16
Move toward the lockers

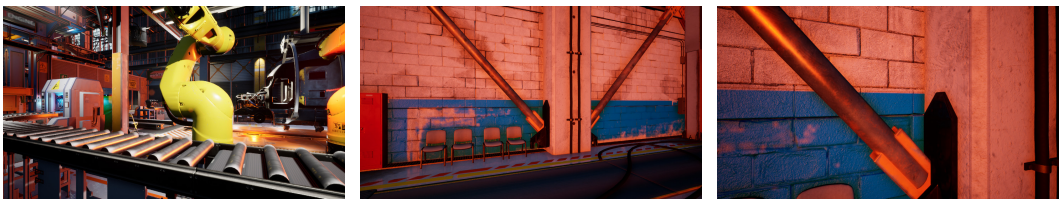
Step 17
At the locker doors



Step 18
Move through the doors

Step 19
Turn to look back out

Step 20
Exit the locker region



Step 30
Continue the inspection

Step 37
Inspect another boundary

Step 40
Final observation

Figure 9: Selected observations while inspecting the solid lockers, ordered left to right and top to bottom.

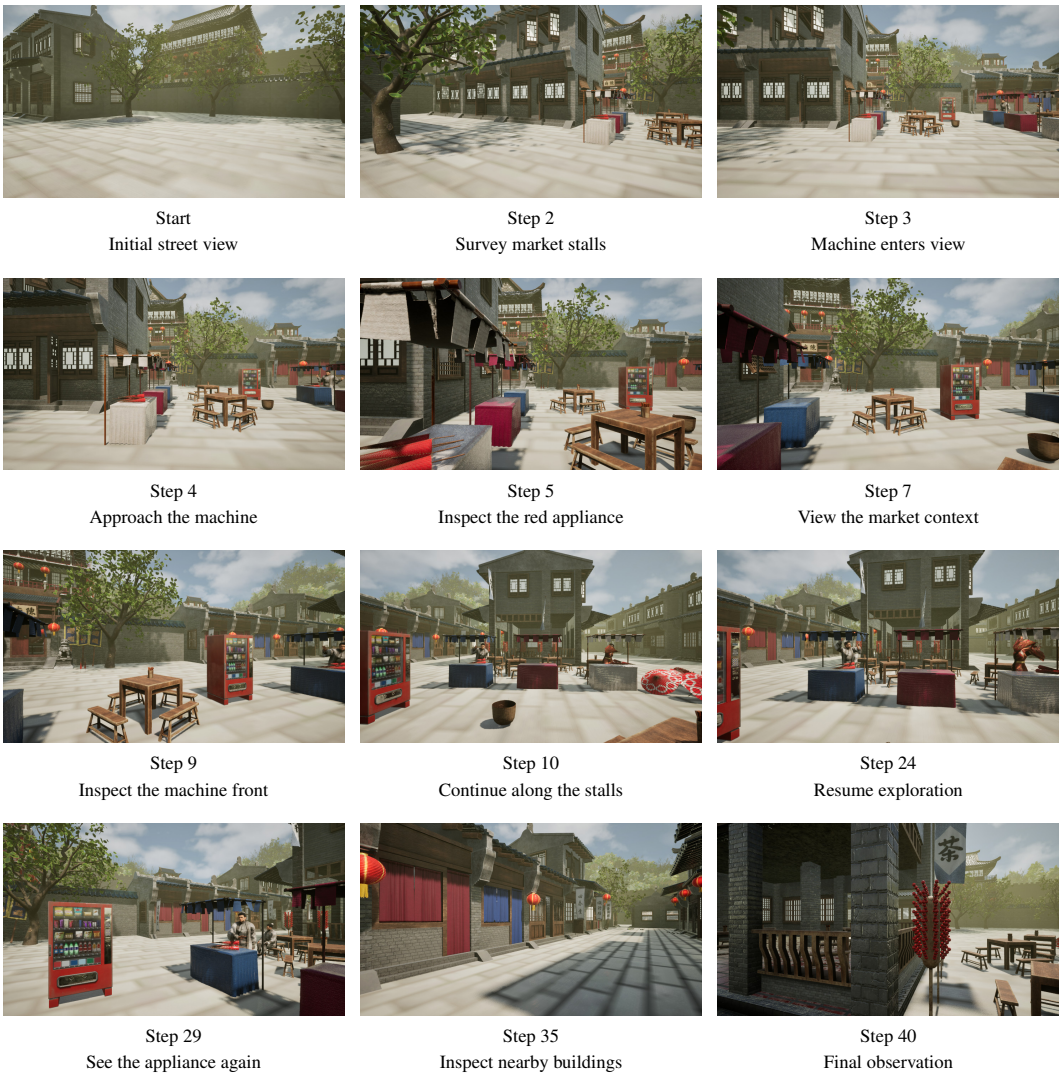
Target anomaly. A visibly solid locker fails to block the player, allowing entry into its body.

Auditor report. The auditor reports that the red lockers lack collision, citing the approach in Steps 15–17, entry through the doors at Step 18, and the reverse view from inside at Step 19.

Interpretation. The report is correct: the entry and reverse views show traversal through a visibly solid surface, matching the target collision failure. The action log also records a one-metre movement at Step 18 without a collision block.

Success: A Modern Appliance In An Ancient Market

Semantic consistency | 40-action budget



Start
Initial street view

Step 2
Survey market stalls

Step 3
Machine enters view

Step 4
Approach the machine

Step 5
Inspect the red appliance

Step 7
View the market context

Step 9
Inspect the machine front

Step 10
Continue along the stalls

Step 24
Resume exploration

Step 29
See the appliance again

Step 35
Inspect nearby buildings

Step 40
Final observation

Figure 10: Selected observations while inspecting the market appliance, ordered left to right and top to bottom.

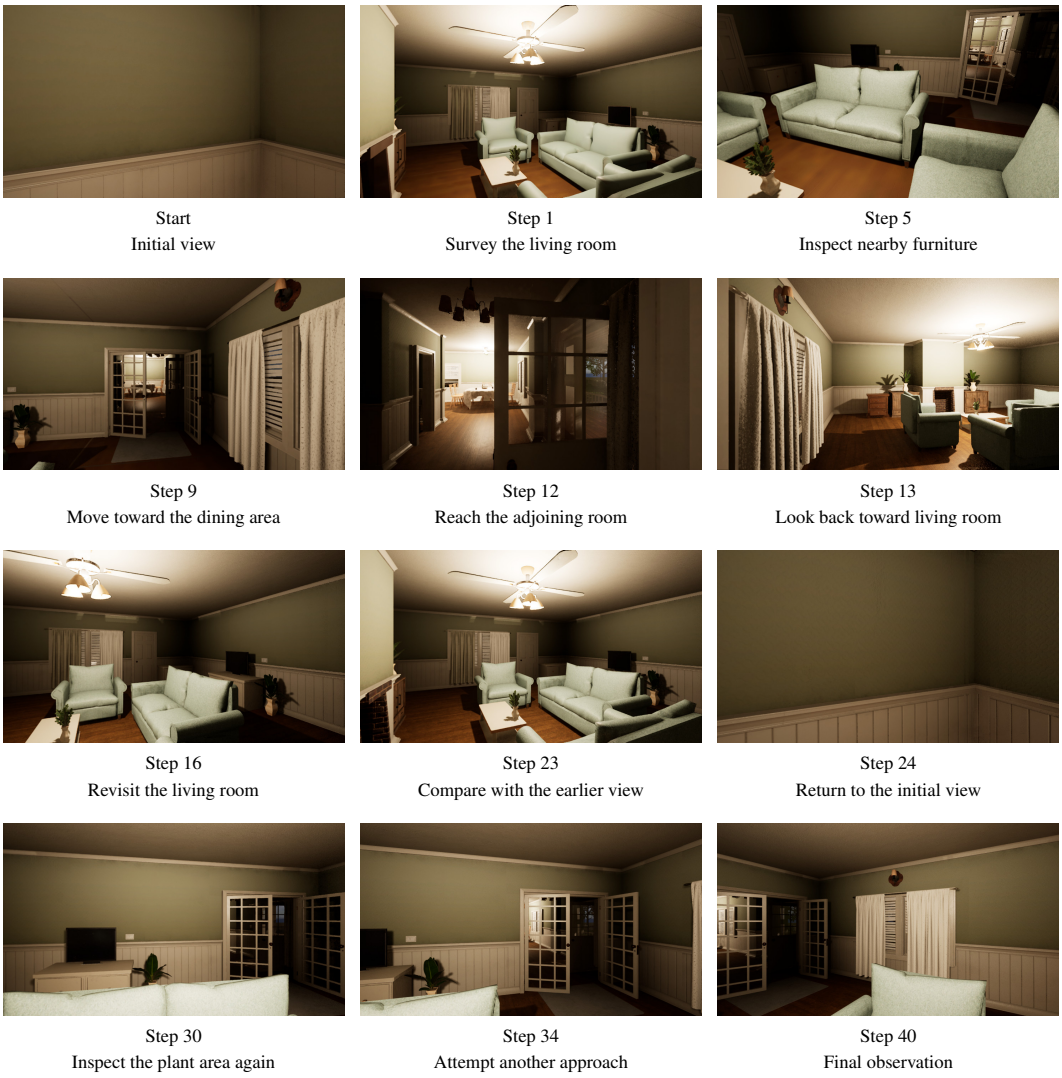
Target anomaly. A modern vending machine appears among the stalls of an ancient Chinese market.

Auditor report. The auditor identifies a modern beverage vending machine that conflicts with the historical setting. Its evidence includes Steps 3–5, 7, and 9, which show the appliance and the surrounding market.

Interpretation. The report is correct: the modern appliance is inconsistent with the ancient market setting, matching the annotated target. The cited views establish both the object's identity and its scene context.

Failure: Missing Before-and-After Evidence

Temporal consistency | 40-action budget



Start
Initial view

Step 1
Survey the living room

Step 5
Inspect nearby furniture

Step 9
Move toward the dining area

Step 12
Reach the adjoining room

Step 13
Look back toward living room

Step 16
Revisit the living room

Step 23
Compare with the earlier view

Step 24
Return to the initial view

Step 30
Inspect the plant area again

Step 34
Attempt another approach

Step 40
Final observation

Figure 11: Selected observations while inspecting the disappearing plant, ordered left to right and top to bottom.

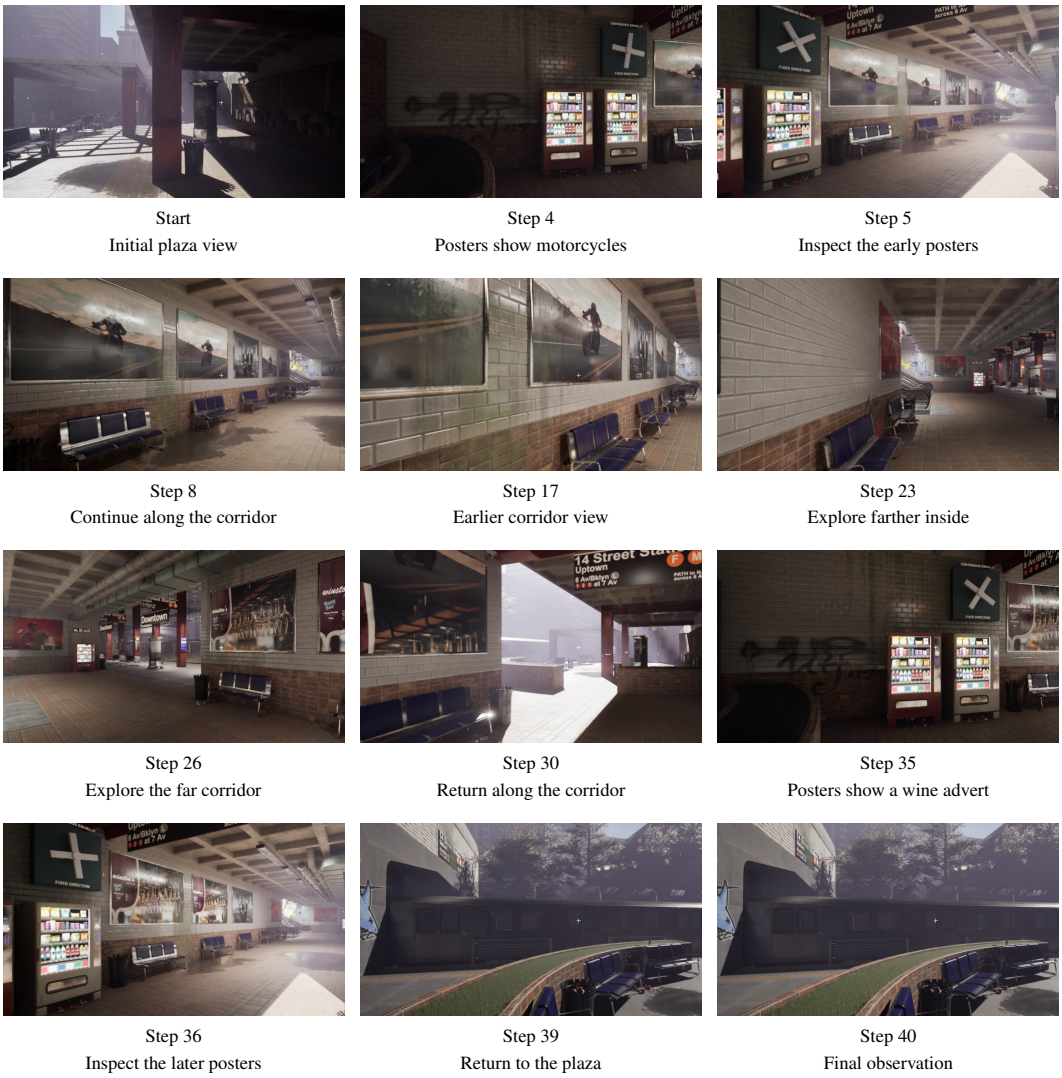
Target anomaly. A floor plant disappears after the auditor approaches it and walks away; the rubric specifies moving more than 3.1 metres from the target.

Auditor report. The auditor reports no anomalies, comparing Step 1 with Step 23 and Start with Step 24 after leaving for the dining area and returning. It concludes that the objects remained consistent.

Interpretation. The auditor revisits the room but fails to obtain a clear before-and-after comparison of the target plant. Its comparisons of broader room views do not establish the plant's disappearance, and it reports no anomalies. This case illustrates a failure to gather the observations needed to identify a temporal anomaly.

Failure: Mistaking Advertisement Cycling For An Anomaly

Temporal consistency | 40-action budget



Start
Initial plaza view

Step 4
Posters show motorcycles

Step 5
Inspect the early posters

Step 8
Continue along the corridor

Step 17
Earlier corridor view

Step 23
Explore farther inside

Step 26
Explore the far corridor

Step 30
Return along the corridor

Step 35
Posters show a wine advert

Step 36
Inspect the later posters

Step 39
Return to the plaza

Step 40
Final observation

Figure 12: Selected observations while inspecting the wall posters, ordered left to right and top to bottom.

Target anomaly. One wall poster disappears after a short time without being touched; the rubric requests observing the two posters for about 15 seconds.

Auditor report. The auditor reports that two framed advertisements change from motorcycle imagery to a pub advertisement after leaving and returning. It cites the earlier views at Steps 4–5 and the later views at Steps 35–36 as evidence.

Interpretation. The auditor mistakes normal advertisement cycling for an anomaly. It compares changing content on the advertising displays rather than establishing the disappearance of the target poster. This produces a false positive while leaving the actual anomaly unidentified: detecting a visual change is insufficient without judging whether that change is expected.

Failure: Misidentifying The Anomalous Object

Static physics | 40-action budget



Start
Initial plaza view

Step 11
Explore the lower plaza

Step 13
Look across the concourse

Step 15
Approach the seating area

Step 17
Inspect the advertising column

Step 18
Look down at its base

Step 21
Move closer to the base

Step 25
View the nearby seating

Step 27
Recheck the column base

Step 35
Inspect the seating area

Step 37
View a nearby bench

Step 40
Final observation

Figure 13: Selected observations while inspecting the floating bench, ordered left to right and top to bottom.

Target anomaly. An entire waiting bench floats approximately 60 centimetres above its support surface.

Auditor report. The auditor reports a floating cylindrical advertising column, citing views of its base in Steps 17–18, 21, and 27. It states that the other objects, including benches, rest properly on their surfaces.

Interpretation. The auditor misidentifies the advertising column as a floating object while overlooking the anomalous bench. Despite repeatedly inspecting the column base, it incorrectly reports the column and concludes that the benches are grounded. This case illustrates a false positive on the wrong object alongside a missed target anomaly.