SPATIALACT: PROBING THE GAP BETWEEN STATIC SPATIAL REASONING AND CLOSED-LOOP ACTION OF VLM AGENTS IN 3D SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

Humans exhibit remarkable spatial intelligence, enabling them to perceive spatial structure, reason about their surroundings, and adapt effectively as environments change. As frontier models such as GPT-6 Astra continue to advance, an important question arises as to how far current vision-language models (VLMs) have progressed from classical spatial understanding toward more complex interactive spatial intelligence. To study this transition, we introduce *SpatialAct*, a simulator-grounded benchmark for evaluating VLMs from static spatial reasoning to closed-loop action in 3D scenes. SpatialAct adopts Multi-turn Interactive Refinement as its primary task and connects foundational spatial abilities and single-step correction within a hierarchical evaluation framework. Experiments across leading proprietary and open-source VLMs reveal a substantial performance gap as tasks progress from isolated spatial judgments to complex multi-turn interaction. While current VLMs perform relatively well on static spatial tasks, their performance drops markedly in closed-loop refinement, where even GPT-6 Astra achieves a Repair Rate of 0.681 and a Scene Success Rate of 0.332, compared with 0.911 and 0.763 for human participants. We further investigate the sources of this cross-task capability gap by decomposing the refinement loop into spatial reasoning, corrective action, and spatial state tracking, revealing distinct limitations that emerge during interaction. Finally, we explore external support for bridging this gap and find that feedback and memory can partially compensate for possible limitations in current VLMs' internal spatial representations, particularly in anticipating action consequences and updating spatial states over time.

1 INTRODUCTION

The emergence of increasingly capable foundation models such as GPT-6 Astra (OpenAI, 2026b) is expanding the potential of agents in 3D environments and embodied systems, raising expectations for the spatial intelligence of vision-language models (Yin et al., 2026; Wang et al., 2025b; Zhao et al., 2025; Zhang et al., 2025b). These applications require models not only to interpret spatial information, but also to accomplish complex tasks as the environment's spatial state evolves through interaction. As agents take on increasingly demanding spatial tasks, an important question is whether progress in classical spatial understanding extends to complex interactive settings. For example, identifying whether a chair is to the left of a table or which object is closest requires a static spatial judgment, whereas rearranging a room requires an agent to decide what to modify, account for how each edit changes surrounding spatial relations, and repeatedly reassess the evolving scene. Current VLMs have made substantial progress on static spatial tasks (Huang et al., 2025b; Xu et al., 2025; Wang et al., 2025c;d), but whether such competence extends to reliable closed-loop interaction remains unclear. This motivates a systematic examination of what capabilities are required for basic spatial competence to support interactive spatial intelligence.

Recent spatial benchmarks have progressively expanded the evaluation of VLMs across different aspects of spatial intelligence. Prior works have studied basic spatial cognition (Huang et al., 2025b; Li et al., 2024b; Wang et al., 2024; Xu et al., 2025), 2D and 3D spatial reasoning (Wang et al., 2025c;d; Zhang et al., 2025c), multi-view understanding (Yang et al., 2025; Yin et al., 2025), and embodied spatial tasks (Majumdar et al., 2024; Wang et al., 2025b; Zhang et al., 2025b; 2026b). However, most

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.

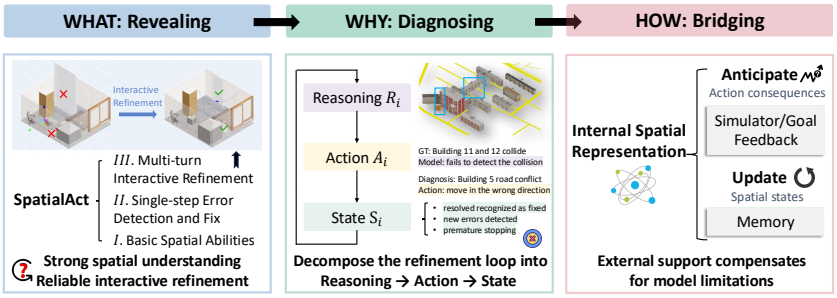


Figure 1: Overview of the SpatialAct framework, illustrating how it reveals the gap, diagnoses refinement failures, and explores external support for improving performance.

existing evaluations still focus on specific spatial capabilities and treat models primarily as observers: models receive visual observations and produce answers, while their outputs do not directly alter the environment that must be reasoned about later. 3D scene generation and editing benchmarks further evaluate whether models can construct or modify environments, such as generating 3D layouts (Yang et al., 2024; Gao et al., 2026) or performing target-driven edits (Yin et al., 2026). However, these settings typically assume a predefined generation goal or target configuration rather than requiring models to diagnose and refine an existing scene through iterative interaction. As a result, existing evaluations have not yet directly characterized the capability transition from static spatial reasoning to reliable closed-loop action.

Motivated by this open question, SpatialAct is organized around three questions: **what** capability gap emerges, **why** it arises, and **how** it can be bridged. For 'what', SpatialAct centers on Multi-turn Interactive Refinement and organizes a three-level hierarchy in which basic spatial abilities and single-step correction jointly characterize foundational spatial competence, while multi-turn refinement evaluates open-ended, multi-error correction under successive feedback. We adopt 3D scene repair because spatial validity, including collision avoidance, boundary consistency, and plausible orientation, is fundamental for usable layouts (Yang et al., 2024; Sun et al., 2025a; Huang et al., 2025a; Wang et al., 2026). This hierarchy reveals a substantial capability drop toward closed-loop interaction, where even GPT-6 Astra achieves only a Repair Rate of 0.681 and a Scene Success Rate of 0.332, compared with 0.911 and 0.763 for humans. For 'why', we decompose the refinement loop into spatial reasoning, corrective action, and spatial state tracking, and diagnose where failures arise and how they manifest. Our analysis shows distinct limitations across all three stages, including unreliable reasoning over evolving scene states, ineffective corrective actions, and difficulty maintaining persistent spatial state. For example, even after correct diagnosis, only 41.58% of repair attempts resolve the target error. For 'how', we explore external support for anticipating action consequences and updating spatial states. Simulator and goal feedback, together with memory, partially compensate for limitations in internal spatial representations, pointing to these two capabilities as important for closed-loop interaction.

The main contributions of our work are summarized as follows:

- We introduce *SpatialAct*, a simulator-grounded benchmark for evaluating the transition from static spatial competence to reliable closed-loop action in 3D environments. It connects Basic Spatial Ability, Single-step Error Detection and Fix, and Multi-turn Interactive Refinement within a unified evaluation framework.
- We conduct comprehensive evaluations of current proprietary and open-source VLMs, revealing a substantial gap between strong performance on isolated spatial tasks and reliable interactive refinement. Even GPT-6 Astra, the strongest evaluated model, remains substantially below human performance in multi-turn scene refinement, highlighting the limitations of current spatial agents.
- We establish a systematic diagnostic framework to analyze the gap between isolated spatial competence and interactive refinement. By decomposing the refinement process into spatial reasoning, corrective action, and spatial state tracking, together with controlled interventions and trajectory-level analysis, we reveal distinct challenges that arise during multi-turn interaction.
- We explore external support mechanisms for bridging the identified gap. Our analysis shows that feedback and memory can partially compensate for limitations in internal spatial representations, highlighting the importance of predictive and state-updating capabilities for closed-loop interaction. Our codes and datasets are open-sourced via `https://anonymous.4open.science/r/SpatialAct-7644/`.

Table 1: Comparison of existing spatial intelligence benchmarks across various evaluation aspects.

Benchmark	Dimension	Modality	Scenario	Instances	Eval Type	Multi-view	Multi-turn	Env. Feed.	Primary Goal
SpatialEval (Wang et al., 2024)	2D	Image	Mix	4.6k	MCQ	✗	✗	✗	Understanding
CoreCognition (Li et al., 2024b)	2D	Mix	Mix	1.5k	MCQ	✗	✗	✗	Understanding
BSA (Xu et al., 2025)	Mix	Image	Classic	312	MCQ	✗	✗	✗	Understanding
Spatial457 (Wang et al., 2025d)	Mix	Image	Outdoor	23.7k	Mix	✗	✗	✗	Understanding
VIS-Bench (Yang et al., 2025)	3D	Video	Indoor	5k	Mix	✓	✗	✗	Understanding
MINDCUBE (Yin et al., 2025)	3D	Image	Mix	21.1k	MCQ	✓	✗	✗	Understanding
CityCube (Xu et al., 2026)	3D	Image	Outdoor	5.0k	MCQ	✓	✗	✗	Understanding
OpenEQA (Majumdar et al., 2024)	3D	Video	Indoor	1.6k	Open-ended	✓	✗	✗	Exploration
ToS (Zhang et al., 2026b)	3D	Image	Indoor	2.7k	Open-ended	✓	✓	✗	Exploration
MetaVQA (Wang et al., 2025b)	3D	Image	Outdoor	9.7k	MCQ	✗	✗	✓	Decision Making
Holodeck (Yang et al., 2024)	3D	Text	Indoor	290	Human	✗	✗	✗	Generation
3DCodeBench (Gao et al., 2026)	3D	Image/Text	Object	26k	Mix	✓	✓	✓	Generation
BlenderBench (Yin et al., 2026)	3D	Image	Indoor	27	Open-ended	✓	✓	✓	Target Editing
SpatialAct	3D	Image	Mix	4.3k	Mix	✓	✓	✓	Interactive Refienment

2 METHODS

We introduce *SpatialAct*, a simulator-grounded benchmark for evaluating agentic VLMs on progressive 3D spatial tasks. It covers three layout scenarios including Abstract Geometric, Urban Architectural, and Indoor Scene, with the latter two constructed from 333 scenes.

2.1 BENCHMARK CONSTRUCTION

To enable systematic evaluation of agentic VLM operating in evolving 3D environments, we construct a scene-level dataset. The pipeline, illustrated in Appendix A.2, consists of data collection, quality control, and QA pair generation. Each scene is represented by both a top-view and an isometric-view, enabling models to access complementary spatial information from different perspectives.

Data Collection SpatialAct consists of three types of scenarios: Abstract Geometric, Urban Architectural, and Indoor Scene. The Abstract Geometric layouts are procedurally generated in a controllable manner. Each scene typically contains 5-7 geometric objects, which are randomly sampled from a predefined set of candidates, including cube, cuboid, cylinder, prism, sphere, L-shape, and U-shape, with varying sizes. For more complex tasks such as Mental Rotation, we further constrain each scene to contain at least one complex L-shaped or U-shaped object, ensuring sufficient structural complexity for spatial reasoning. The Urban Architectural layouts are derived from UrbanWorld2.0 (Wang et al., 2025a), a multimodal agent framework for reality-aligned 3D world generation. We use both its white-box building models and textured building models to construct mixed urban scenes, while restricting each scene to contain no more than 20 buildings. The Indoor Scene layouts are built from InternScenes (Zhong et al., 2025), which provides diverse indoor environments with movable furniture meshes. Considering the higher visual and structural complexity of indoor environments, we select scenes containing 5–15 objects for benchmark construction.

Quality Control For Abstract Geometric scenes, the layouts are generated procedurally under predefined constraints, and therefore no additional filtering is required. For Urban Architectural and Indoor Scene layouts, although the original scene-generation pipelines already provide relatively well-structured scenes, our benchmark focuses on error patterns in spatial configurations. We therefore first apply programmatic cleaning to remove existing abnormal cases from the original scenes. We then conduct manual filtering to examine abnormal element sizes, the plausibility of spatial arrangements, and potential error patterns. For Indoor Scene layouts, we additionally check semantic consistency, as indoor objects carry rich semantic categories. These inspections ensure the quality and reliability of the final benchmark scenes.

Task Design and QA Pairs Generation SpatialAct follows a hierarchical task design centered on Multi-turn Interactive Refinement, the most challenging setting in our benchmark. In this task, we first inject layout errors of varying difficulty and correction steps into clean, error-free scenes obtained from the previous filtering stage. Models are required to iteratively repair abnormal errors in a scene through multi-turn action-based interaction. A representative indoor-scene prompt and response format can be found in Appendix A.1.7. To facilitate analysis of the underlying causes of model performance, we further design two levels of diagnostic tasks. First, Single-step Error Detection and Fix decomposes the interactive refinement process into a one-step setting, focusing on whether models can detect a spatial error and choose the corresponding correction. Second, we introduce five basic spatial ability tasks: Object Meaning, Spatial Relation, Spatial Orientation, Mental Rotation, and Spatial Visualization. These dimensions are adapted from prior work on basic spatial abilities (Xu et al., 2025), which identifies five key dimensions based on psychometric theories. While the original

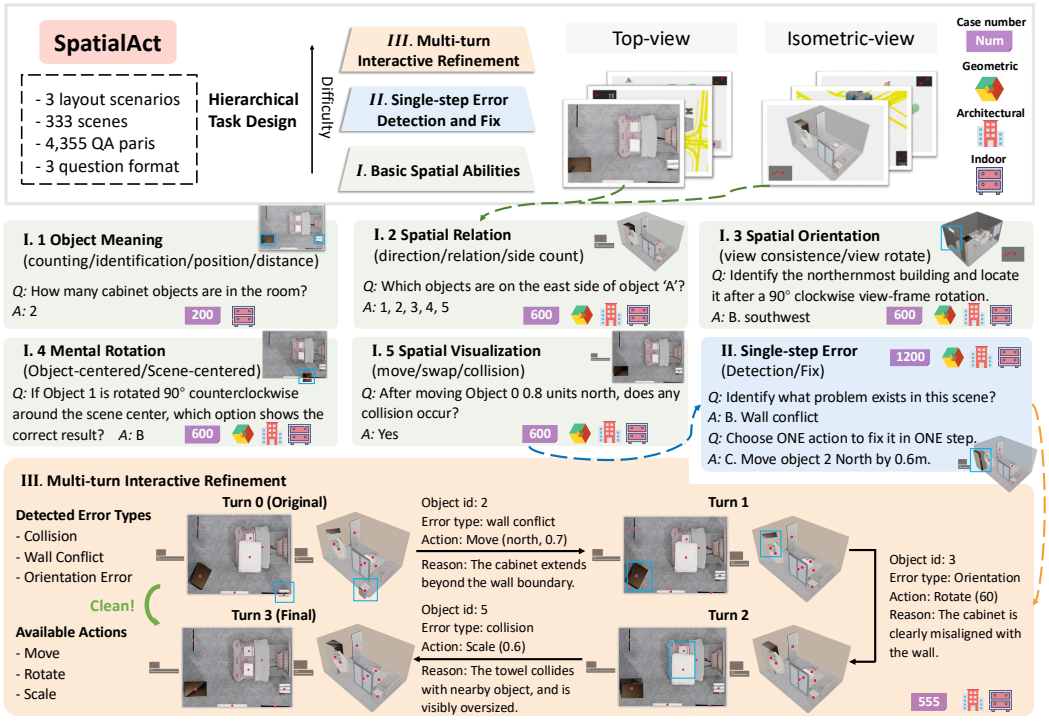


Figure 2: Overview of the SpatialAct benchmark, illustrating its hierarchical task design, task composition, and multi-turn interactive refinement process.

tasks are mainly designed around classical geometry and cube-like scenarios, we reinterpret their theoretical meanings and redesign the questions as foundational components underlying Multi-turn Interactive Refinement. Each task category contains multiple QA subtypes, each emphasizing aspects of spatial cognition. Detailed QA subtype definitions are provided in the Appendix A.1.1. Depending on the characteristics of each task, QA pairs are formulated in three formats: open-ended questions, multiple-choice questions (MCQs), and multi-turn feedback-based interaction.

2.2 Environment and Interaction Design

To study how VLM agents operate in evolving 3D environments, we design Multi-turn Interactive Refinement, a closed-loop task where models iteratively interact with the simulator, receive feedback from action-induced scene changes, and update their decisions. Unlike static QA settings, this task evaluates whether models can maintain consistent spatial states over long-horizon interaction.

Multi-turn Iterative Rendering Workflow In the Multi-turn Interactive Refinement task, VLMs are required to detect and correct abnormal spatial configurations through iterative actions that progressively alter the scene state. These error types are designed to reflect common failure modes in 3D layout construction (Sun et al., 2025a; Feng et al., 2026; Wang et al., 2026), where objects or buildings must avoid physical conflicts, stay within valid spatial boundaries, and maintain plausible orientations. For Urban Architectural scenes, the model is asked to identify three types of building-related errors: collision, road conflict, and orientation error. For Indoor Scene layouts, the model focuses on three types of object-related errors: collision, wall conflict, and orientation error. At each turn, the model can manipulate the current scene using three types of actions: `move(direction, number)`, `rotate(degree)`, and `scale(number)`. Each interaction starts by providing the VLM with the top-view and isometric-view renderings of an original erroneous scene. The model then determines what spatial errors exist and outputs corrective action commands. These commands are parsed by and executed in the simulator, after which the updated scene is rendered again from both the top and isometric views. In the next turn, the VLM receives the newly rendered views together with the original top-view and isometric-view images, the last-turn action command, and the accumulated previous action history. This iterative process continues until the VLM judges that all errors in the scene have been fixed or the predefined maximum number of turns is reached.

Human Evaluation Platform To establish a human reference for the Multi-turn Interactive Refinement task, we develop a web-based 3D evaluation platform that enables participants to iteratively

Table 2: Main results on the SpatialAct. Rep. Rate denotes Repair Rate, Succ. Rate denotes Scene Success Rate, Eff. Turn denotes Effective Repair Turn Ratio, and Tok./Scene denotes Average Completion Tokens per Scene.

Task	**Multi-turn Refinement**					**Single-step Edit**	**Object Meaning**	**Spatial Relation**	**Spatial Orientation**	**Mental Rotation**	**Spatial Visualization**
Metrics	Rep. Rate ↑	Succ. Rate ↑	Eff. Turn ↑	Scene MAU ↑	Tok./ Scene	Acc. ↑	Acc. ↑	Acc. ↑	Acc. ↑	Acc. ↑	Acc. ↑
Proprietary Models											
GLM-5V-Turbo	-0.012	0.035	0.268	0.031	31101	0.452	0.695	0.738	0.617	0.500	0.665
GPT-5.4 mini	0.088	0.009	0.272	0.069	10060	0.595	0.720	0.810	0.616	0.570	0.813
GPT-5.4	0.208	0.038	0.270	0.208	23040	0.664	0.751	0.826	0.625	0.690	0.850
Gemini-3.1 Pro	0.411	0.206	0.320	0.361	22814	0.721	0.770	0.835	0.720	0.628	0.785
GPT-6 Astra	**0.681**	**0.332**	**0.764**	**0.755**	1087	**0.882**	**0.815**	**0.892**	0.623	**0.938**	**0.888**
Open-source Models											
DeepSeek-V4.1-Flash	0.042	0.033	0.226	0.084	117272	0.488	0.640	0.807	**0.738**	0.573	0.780
Kimi-K2.5	0.032	0.009	0.055	0.003	9211	0.368	0.615	0.670	0.393	0.328	0.473
Qwen3.6-27B	0.035	0.005	0.020	0.005	8565	0.343	0.600	0.668	0.446	0.155	0.581
Qwen3.6-35B-A3B	-0.099	0.016	0.264	-0.053	196313	0.442	0.640	0.788	0.681	0.585	0.700
Qwen3.8-27B	0.073	0.029	0.200	0.100	7815	0.517	0.720	0.857	0.718	0.618	0.845

inspect and correct spatial inconsistencies. The platform provides the same scene observations and transformation space as the model setting, including fixed top-down and isometric views, labeled object selection, and position, rotation, and scale adjustments. Unlike models that generate transformations through the textual action grammar, human participants can directly manipulate objects through an interactive interface. The interface design and human evaluation protocols are described in Appendix A.3.

3 Experiments

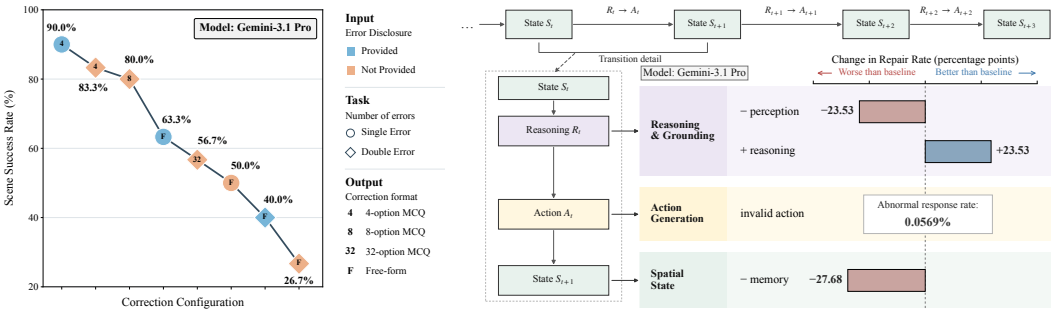
3.1 Experimental Setup

We comprehensively evaluate 10 general-purpose VLMs, covering diverse model scales and model families. For proprietary models, we include GPT-5.4 mini, GPT-5.4 (OpenAI, 2026a), GLM-5V-Turbo (Hong et al., 2026), Gemini-3.1 Pro (DeepMind, 2026), and GPT-6 Astra (OpenAI, 2026b). For open-source models, we evaluate DeepSeek-V4.1-Flash (DeepSeek-AI, 2026), Kimi-K2.5 (Team & et al., 2026), Qwen3.6-27B (Qwen Team, 2026a), Qwen3.6-35B-A3B (Qwen Team, 2026b), and Qwen3.8-27B (Qwen Team, 2026c). We additionally evaluate domain-specific spatial VLMs in Appendix A.5. During evaluation, we adopt a unified inference setting across all models. For the Multi-turn Interactive Refinement task, we set the maximum number of iterative turns to 30. Since the evaluated models are reasoning models with a context window of approximately 200K tokens, we limit the maximum number of completion tokens for each single-turn response to 8,096. We further examine the effect of different context window sizes on model behavior in Appendix A.12. For basic spatial abilities and single-step editing tasks, we report task accuracy, measuring whether models can correctly judge spatial relations or select appropriate scene modifications. For multi-turn interactive refinement, we evaluate both repair effectiveness and interaction efficiency using Repair Rate, Scene Success Rate, Effective Repair Turn Ratio, Mean Scene MAU, and Average Completion Tokens per Scene. Detailed metric definitions are provided in Appendix A.4.

3.2 Multi-turn Refinement Performance and Bottleneck Diagnosis

We first focus on Multi-turn Interactive Refinement, the most challenging and primary task in SpatialAct. We analyze how current VLM agents perform under closed-loop spatial interaction and diagnose the factors that limit reliable refinement. We additionally investigate how scene characteristics influence refinement performance under controlled settings in Appendix A.13.

Overall Benchmark Performance: Models and Humans As shown in Table 2, Multi-turn Interactive Refinement exposes substantial differences among current VLMs. Open-source models achieve Repair Rates near zero or negative, while stronger proprietary models perform considerably better. GPT-6 Astra leads with a Repair Rate of 0.681 and a Scene Success Rate of 0.332, while Gemini-3.1 Pro reaches 0.411 and 0.206, respectively. GPT-6 Astra also attains the highest Effective Repair Turn Ratio of 0.764 with an average of 2.71 refinement turns, showing substantially stronger efficiency and stability throughout interaction. Despite this advantage, GPT-6 Astra remains well below human performance. Human participants achieve a **Repair Rate of 0.911** and a **Scene Success Rate of 0.763**, demonstrating strong human capability in identifying and correcting spatial inconsistencies through iterative scene manipulation. The substantial performance spread across VLMs and the persistent gap to humans motivate our diagnosis of what still limits current models in



(a) (b)

Figure 3: (a) Correction accuracy under different single-turn task structures, showing the increasing difficulty of open-ended correction. (b) Diagnostic interventions revealing the impact of spatial reasoning & grounding, action generation, and persistent state maintenance on multi-turn refinement.

multi-turn interaction and what distinguishes stronger from weaker agents. Most controlled analysis therefore use Gemini-3.1 Pro as an intermediate-capability model, with meaningful repair ability but substantial room for improvement relative to GPT-6 Astra and humans.

How Task Structure Shapes Correction Difficulty To better understand what makes spatial correction difficult, we conduct single-turn experiments with Gemini-3.1 Pro on 30 mixed-domain scenes and vary three aspects of the task: the information provided in the input, the number of errors to be corrected, and the structure of the output space. Figure 3a organizes representative configurations. Three consistent patterns emerge. First, providing the error information reduces difficulty. Under free-form correction, success increases from 50.0% to 63.3% for single-error cases and from 26.7% to 40.0% for double-error cases when the error is disclosed. Second, multiple errors substantially increase the reasoning burden. With free-form output, moving from one error to two reduces success from 63.3% to 40.0% when the error is known, and from 50.0% to 26.7% when it is unknown. Third, correction becomes increasingly difficult as the output space expands. For unknown double-error cases, success falls from 83.3% with 4 options to 80.0% with 8 options, 56.7% with 32 options, and 26.7% with free-form generation. These results indicate that correction difficulty compounds across input uncertainty, multi-error reasoning, and open-ended action generation.

Diagnosing the Sources of Failure The previous results show that open-ended multi-turn refinement becomes difficult when multiple capabilities must be jointly composed. To diagnose the underlying failures, we conduct controlled interventions with Gemini-3.1 Pro on a subset of cases. $-$*Perception* removes predefined object grounding and requires the model to infer object identities from visual inputs, $+$*Reasoning* provides oracle diagnoses with object IDs and error types while leaving repair action generation to the model, and $-$*Memory* removes cross-round history, requiring the model to infer the current state from the present observation only. We also analyze invalid actions to test whether failures arise from unusable commands. Detailed intervention settings are in the Appendix A.6. As illustrated in Figure 3b, removing perception support reduces Repair Rate by 23.53 percentage points (pp), while oracle diagnosis improves it by 23.53 pp, highlighting the importance of scene understanding and spatial reasoning. Removing cross-round memory causes a 27.68 pp drop, showing that maintaining persistent spatial state is critical for long-horizon refinement. Here, *spatial state* broadly refers to the evolving spatial configuration of an environment at a given moment. In contrast, invalid actions are extremely rare, suggesting that the main action bottleneck lies beyond command validity. Together, the diagnostic analysis identifies three interacting capability dimensions underlying multi-turn refinement failures: **reliable spatial reasoning**, **effective action execution**, and **persistent spatial state maintenance**. Qualitative cases are provided in Appendix A.11.

3.3 From Basic Spatial Skills to Interactive Spatial Reasoning

Multi-turn refinement builds upon fundamental spatial abilities and requires models to update their understanding as the environment changes. We therefore characterize isolated spatial competence and analyze how reasoning evolves when models must repeatedly interpret an evolving scene.

Foundational and Single-step Spatial Reasoning In Table 2, we evaluate Single-step Error Detection and Fix together with five basic spatial ability tasks as decomposed components of Multi-turn Interactive Refinement. Their predictive relationships across models are further analyzed in Appendix A.12. On these isolated tasks, GPT-6 Astra and Gemini-3.1 Pro generally perform best, while model differences remain much smaller than in multi-turn refinement. GPT-6 Astra reaches

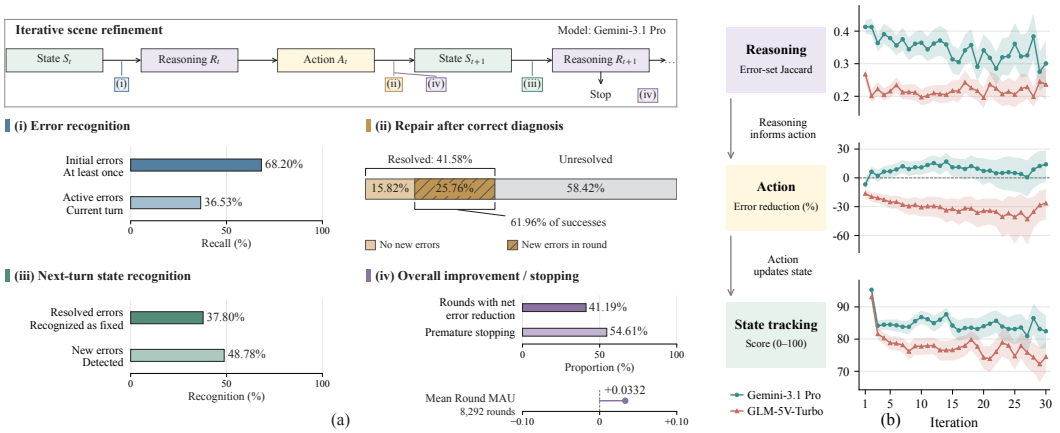


Figure 4: (a) Capability-level analysis of Gemini-3.1 Pro across the full evaluation set. iterative refinement process is decomposed into error recognition, repair after correct diagnosis, next-turn state recognition, and overall refinement effectiveness. (b) Dynamics of reasoning, action progress, and spatial state tracking across interaction rounds for Gemini-3.1 Pro and GLM-5V-Turbo.

0.882 on Single-step Error Detection and Fix and leads four of the five foundational tasks. Spatial Orientation is the exception, where DeepSeek-V4.1-Flash reaches 0.738 versus 0.623 for GPT-6 Astra; further inspection shows that GPT-6 Astra struggles particularly with joint camera-view and coordinate-frame rotations. The relatively strong performance on these tasks suggests that basic spatial competence alone cannot fully explain the multi-turn performance gap, motivating further analysis of reasoning over evolving scene states.

Reasoning over Evolving Scene States Multi-turn refinement requires a model not only to recognize spatial errors, but also to continuously reassess the scene after each action. In Figure 4a, we evaluate Gemini-3.1 Pro using trajectory-level cumulative recall and turn-level state recognition. Across trajectories, Gemini-3.1 Pro identifies 68.20% of initial errors at least once while active, but its turn-level recall of currently active errors is only 36.53%. Following an action, the model explicitly recognizes 37.80% of actually resolved errors as fixed in the next turn and detects 48.78% of newly introduced errors in the next turn. This incomplete state understanding is also reflected in termination behavior. Gemini-3.1 Pro prematurely stops in 54.61% of measurable trajectories, producing no actionable repair while simulator-detected errors remain and before reaching the predefined turn limit. Taken together, the above results reveal that current VLMs can identify spatial errors, but struggle to maintain an accurate and updated understanding of the evolving scene state after actions.

3.4 Beyond Action Generation: Effective Spatial Intervention

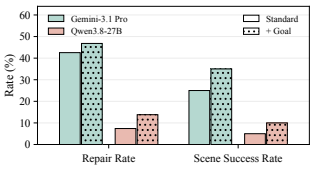
Correctly understanding a spatial error is only the first step toward successful refinement. A capable spatial agent should also maintain an internal world model to predict how its actions will transform the environment and guide subsequent decisions.

Translating Correct Diagnosis into Effective Repair Correct spatial diagnosis does not guarantee effective repair. Using the same Gemini-3.1 Pro trajectories, we evaluate repair attempts targeting correctly identified object–error-type units, as summarized in Figure 4a. The target error is resolved after the corresponding action round in only 41.58% of these attempts. Moreover, 61.96% of successful target repairs occur in rounds that introduce at least one new error unit, leaving only 15.82% of all correctly diagnosed attempts both successful and free of new errors. At the turn level, among rounds containing at least one correctly diagnosed repair attempt, only 41.19% remove more error units than they introduce. These results show that even correct diagnosis often fails to translate into a locally successful and globally safe edit. We further quantify geometric change using Round Marginal Action Utility (Round MAU; Appendix A.7), which captures signed changes in error severity beyond discrete counts. Gemini-3.1 Pro achieves a mean Round MAU of only $+0.0332$ across 8,292 actionable rounds, indicating limited net geometric improvement on average. Together, these results highlight a substantial gap between identifying what is wrong and executing edits that reliably improve the scene.

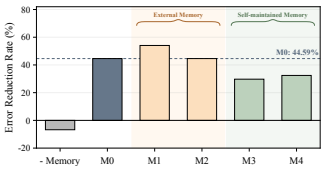
Probing Spatial World Modeling through Simulator-Guided Lookahead Reliable action execution requires anticipating how edits transform the scene state. We hypothesize that current VLM

Model	Base	Sim. Feedback	Δ
Gemini-3.1 Pro	0.433	0.711	+0.278
Best-of-3	0.433	0.629	+0.196
GPT-6 Astra	0.639	**0.742**	+0.103
GPT-5.4	0.206	0.289	+0.083
GPT-5.4 mini	0.103	0.175	+0.072

(a)



(b)



(c)

Figure 5: (a) Effect of simulator feedback on Repair Rate. (b) Goal-status feedback improves refinement performance of both models. (c) Comparison of different memory designs.

agents may lack a sufficiently reliable internal world model for predicting such transitions, and introduce simulator-guided trial-and-refinement with external consequence feedback from Blender. Inspired by human interactive editing, where users iteratively modify scenes and inspect outcomes before committing changes, the model receives up to three tentative trials per repair round, each executed from the same round-start state, and revises its plan based on the resulting scene before committing only the final action. We also compare against Best-of-3, which generates three candidates without feedback and selects the best outcome, to isolate the benefit of feedback from additional sampling. Figure 5a summarizes results on the matched 20-scene set. Simulator-guided trial-and-refinement improves Repair Rate across all four models by 7.2–27.8 percentage points. Gemini-3.1 Pro reaches 0.711, exceeding Best-of-3 (0.629), suggesting that consequence feedback provides additional benefits beyond increased sampling. Interestingly, Gemini-3.1 Pro substantially underperforms GPT-6 Astra in the original setting, but approaches comparable performance when both equipped with simulator-guided feedback (0.711 vs. 0.742). In addition, GPT-6 Astra achieves strong performance on Mental Rotation and Spatial Visualization in Table 2, where models are required to predict spatial configurations after hypothetical transformations. This provides evidence that GPT-6 Astra may have stronger implicit world modeling capability for anticipating action-induced spatial transitions, whereas simulator feedback primarily compensates for this limitation in weaker models. The observed gains demonstrate that external simulator feedback can partially bridge the gap between current VLMs' internal spatial transition prediction and reliable action consequence modeling.

Goal Completion Feedback for Iterative Refinement Motivated by recent goal-oriented agent systems and the frequent premature stopping observed in SpatialAct, we introduce goal-completion feedback at the output of each interaction round to guide subsequent refinement. Specifically, this feedback takes the form of a binary signal indicating whether residual geometric errors remain, without revealing their locations, types, or corrective actions; the original refinement procedure and stopping criteria remain unchanged. We evaluate 20 cases with Gemini-3.1 Pro and Qwen3.8-27B, representing relatively stronger and weaker models. As shown in Figure 5b, both models improve in Repair Rate and Scene Success Rate, with the latter increasing by 5–10 percentage points. This shows that completion feedback is helpful for subsequent refinement, but remains far from sufficient for successful scene repair. The exact feedback wording is given in Appendix A.8. Beyond such round-level feedback, reliable long-horizon refinement also depends on maintaining a coherent spatial state across interaction, which we examine next.

3.5 Maintaining Persistent Spatial State across Interaction

Interactive refinement requires not only effective individual actions, but also a coherent representation of the evolving environment. We examine spatial world modeling dynamics across interaction rounds and how different memory designs support spatial state maintenance.

Capability Dynamics across Interaction Rounds We examine how reasoning, repair progress, and spatial state quality evolve over multi-turn interaction for Gemini-3.1 Pro and GLM-5V-Turbo on the full evaluation set. At each round, we evaluate matched scenes reached by both models, pooling the three scene settings. Reasoning is measured by current-round error-set Jaccard, repair progress by cumulative error-count reduction from the initial scene, and spatial state tracking quality by a GPT-5.6-sol-based score. Full metric definitions and fixed-cohort analysis are provided in the Appendix A.9. Figure 4b reveals three patterns. First, Gemini-3.1 Pro consistently achieves higher error-recognition scores than GLM-5V-Turbo, although its scores generally decline rather than improve over successive rounds. Second, the models diverge in cumulative repair progress. Gemini-3.1 Pro achieves positive mean error reduction after the first round, with non-monotonic gains, whereas GLM-5V-Turbo remains below zero, indicating net error accumulation relative to the initial scene. Third, spatial-state scores drop sharply during the early rounds and fluctuate thereafter, with Gemini-3.1 Pro maintaining higher scores. Overall, these dynamics show that additional interaction

does not uniformly improve all three dimensions. Instead, performance remains unstable across rounds, revealing persistent weaknesses in sustaining coherent spatial understanding and effective correction over long-horizon interaction.

Exploring Memory Designs for Persistent Spatial State We further explore how different memory designs support persistent spatial state using Gemini-3.1 Pro on 15 mixed-domain scenes. We compare the default benchmark setting (M0) with four memory designs: an external temporal 2D map (M1), an external metric 3D state (M2), a self-updated cognitive map (M3), and a self-summarized action–outcome memory (M4), together with a memoryless variant. Full memory representations and implementation details are provided in the Appendix A.10. Figure 5c shows that memory design has a substantial effect on refinement performance. Removing cross-round memory leads to a negative Repair Rate, while the external temporal 2D map (M1) achieves the best performance at 54.05%, exceeding the default benchmark setting (M0). Other memory variants provide limited gains or even degrade performance, with the self-maintained designs performing particularly poorly. This contrast suggests that current VLMs still struggle to autonomously update and preserve a reliable internal spatial world model across interaction, whereas externally maintained representations can provide more stable support.

4 Related Work

3D Spatial Intelligence Evaluation Existing spatial intelligence benchmarks evaluate VLMs from increasingly diverse perspectives, ranging from fundamental spatial perception and reasoning to embodied tasks in 3D environments. Several benchmarks (Wang et al., 2025d; Zhang et al., 2025c; Wang et al., 2025c; Ramakrishnan et al., 2024; Kanade & Ganu, 2026; Liu et al., 2025; Yu et al., 2025; Xiao et al., 2026; Rahmanzadehgervi et al., 2024), such as Spatial-DISE (Huang et al., 2025b) and BSA (Xu et al., 2025), organize classical spatial tasks into hierarchical evaluation suites to systematically assess fundamental spatial abilities. More recent evaluations broaden this scope toward more diverse environments, spatial scales, and visual observations (Wang et al., 2024; Li et al., 2024b; Yang et al., 2025; Yin et al., 2025; Zhao et al., 2025; Li et al., 2025; Sun et al., 2025b). For example, CityCube (Xu et al., 2026) incorporates observations from multiple viewpoints and orientations. A growing line of work has begun to incorporate action-oriented or embodied settings (Wang et al., 2025b; Zhang et al., 2025b; Li et al., 2024a; Zhang et al., 2026b), extending spatial evaluation beyond static understanding. However, existing benchmarks typically evaluate individual tasks or capabilities, making it difficult to understand how models generalize from spatial perception to interactive problem solving. SpatialAct addresses this gap by providing a unified evaluation framework spanning Basic Spatial Abilities, Single-step Error Detection and Fix, and Multi-turn Interactive Refinement.

3D Layout Generation and Editing Recent advances in 3D layout modeling have explored layout generation (Feng et al., 2026; Gao et al., 2026) and layout editing (Huang et al., 2025a; Abdelreheem et al., 2025; El Amine Boudjoghra et al., 2025; Weihs et al., 2021). On the generation side, LayoutGPT (Feng et al., 2023) uses Large Language Models (LLMs) to generate structured layouts, while Holodeck (Yang et al., 2024) and LayoutVLM (Sun et al., 2025a) further combine VLM reasoning with spatial constraints or differentiable optimization to improve semantic coherence and physical plausibility. On the editing side, recent works explore language-guided layout editing (Zhen et al., 2026) and evaluate VLM agents on goal-oriented 3D editing tasks (Yin et al., 2026). However, existing works mainly focus on generating or editing 3D layouts toward predefined objectives, rather than evaluating their ability to iteratively diagnose and refine spatial states under feedback. In contrast, SpatialAct introduces a multi-turn spatial refinement setting, where agents iteratively diagnose scene states, apply corrective operations, and improve layouts through interaction feedback.

5 Conclusion

We present *SpatialAct*, a simulator-grounded benchmark for evaluating whether VLMs can translate static spatial reasoning into reliable interactive action in 3D environments. By connecting basic spatial abilities, single-step correction, and multi-turn refinement, SpatialAct reveals a substantial gap between isolated spatial competence and closed-loop interaction. Our analysis shows that current VLMs struggle across the full interactive refinement process, including spatial reasoning over evolving scene states, effective corrective action generation, and coherent spatial state maintenance, with even frontier models remaining below human performance. Furthermore, external feedback and memory can partially compensate for limitations in current VLMs' internal spatial world models, highlighting the importance of predictive and updating capabilities for future spatial agents.

AI USE STATEMENT

In this work, we used generative AI tools for [assist with translation]. We have not used generative AI tools for [generate synthetic data sets, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, interpret results, support qualitative and thematic data analysis], and [help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses] are not applicable to this work. Additionally, we used generative AI tools for [edit a research paper to improve readability]. We have reviewed all AI-assisted work. [All AI-assisted text was carefully checked, critically considered, and verified by the authors]. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the gap between static spatial reasoning and closed-loop action of VLM agents in 3D scenes. Our benchmark is constructed using synthetic and publicly available 3D resources, and does not involve the collection, processing, or release of personal data or sensitive information. We conduct a human evaluation to establish a reference point for interactive spatial refinement performance. Human participants voluntarily performed the refinement task through a web-based 3D interface under the same evaluation objective and transformation space as the model setting. Participants were provided with appropriate compensation for their time and effort. No personally identifiable information was collected, and all human evaluation data were used solely for aggregate research analysis. The proposed benchmark is intended to facilitate research on improving the reliability, interpretability, and transparency of VLM agents in interactive spatial tasks. The benchmark is designed as a controlled research environment for evaluating and diagnosing model capabilities, and we encourage its responsible use for scientific research and development.

REPRODUCIBILITY STATEMENT

Our work aims to ensure transparency and reproducibility through the release of both code and data. All implementation and evaluation code is publicly available at `https://anonymous.4open.science/r/SpatialAct-7644/`. For the data release, the complete dataset (approximately 17 GB) cannot be directly hosted in the anonymous repository due to the file size limitations and release constraints of Anonymous GitHub, where individual files larger than 8 MB are not supported. Therefore, the current repository includes the complete evaluation JSON files, together with a representative subset of images and Blender (`.blend`) scene examples. The full dataset, including all associated images and Blender scene files, will be publicly released upon acceptance of the paper.

REFERENCES

Ahmed Abdelreheem, Filippo Aleotti, Jamie Watson, Zawar Qureshi, Abdelrahman Eldesokey, Peter Wonka, Gabriel Brostow, Sara Vicente, and Guillermo Garcia-Hernando. Placeit3d: Language-guided object placement in real 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6645–6655, 2025.

Zhongang Cai, Ruisi Wang, Chenyang Gu, Fanyi Pu, Junxiang Xu, Yubo Wang, Wanqi Yin, Zhitao Yang, Chen Wei, Tongxi Zhou, et al. Scaling spatial intelligence with multimodal foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7879–7890, 2026.

Google DeepMind. Gemini-3.1 pro. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`, 2026.

DeepSeek-AI. Deepseek-v4.1-flash: Pushing the limits of kv cache compression, 2026.

Mohamed El Amine Boudjoghra, Ivan Laptev, and Angela Dai. Scanedit: Hierarchically-guided functional 3d scan editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27105–27115, 2025.

Yue Fan, Xiaojian Ma, Rongpeng Su, Jun Guo, Rujie Wu, Xi Chen, and Qing Li. Embodied videoagent: Persistent memory from egocentric videos and embodied sensors enables dynamic scene understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6342–6352. IEEE, 2025.

Haoran Feng, Yifan Niu, Zehuan Huang, Yang-Tian Sun, Chunchao Guo, Yuxin Peng, and Lu Sheng. Repurposing 3d generative model for autoregressive layout generation. *arXiv preprint arXiv:2604.16299*, 2026.

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

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

Wenyi Hong, Xiaotao Gu, Ziyang Pan, Zhen Yang, Yuting Wang, Yue Wang, Yuanchang Yue, Yu Wang, Yanling Wang, Yan Wang, et al. Glm-5v-turbo: Toward a native foundation model for multimodal agents. *arXiv preprint arXiv:2604.26752*, 2026.

Ian Huang, Yanan Bao, Karen Truong, Howard Zhou, Cordelia Schmid, Leonidas Guibas, and Alireza Fathi. Fireplace: Geometric refinements of llm common sense reasoning for 3d object placement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13466–13476, 2025a.

Xinmiao Huang, Qisong He, Zhenglin Huang, Boxuan Wang, Zhuoyun Li, Guangliang Cheng, Yi Dong, and Xiaowei Huang. Spatial-dise: A unified benchmark for evaluating spatial reasoning in vision-language models. *arXiv preprint arXiv:2510.13394*, 2025b.

Suchae Jeong, Jaehwi Song, Haeone Lee, Hanna Kim, Jian Kim, Dongjun Lee, Dong Kyu Shin, Changyeon Kim, Dongyoon Hahm, Woogyeol Jin, et al. Learning multi-view spatial reasoning from cross-view relations. *arXiv preprint arXiv:2603.27967*, 2026.

Yuanchen Ju, Yongyuan Liang, Yen-Jen Wang, Gireesh Nandiraju, Yuanliang Ju, Seungjae Jay Lee, Qiao Gu, Elvis Hsieh, Furong Huang, and Koushil Sreenath. Momagraph: State-aware unified scene graphs with vision-language models for embodied task planning. In *International Conference on Learning Representations*, volume 2026, pp. 96361–96386, 2026.

Aditya Sanjiv Kanade and Tanuja Ganu. Do you see me: A multidimensional benchmark for evaluating visual perception in multimodal llms. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7285–7326, 2026.

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, et al. Viewspatial-bench: Evaluating multi-perspective spatial localization in vision-language models. *arXiv preprint arXiv:2505.21500*, 2025.

Haoyang Li, Yang You, Hao Su, and Leonidas Guibas. Physmem: Scaling test-time memory for embodied physical reasoning. *arXiv preprint arXiv:2602.20323*, 2026.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Li, Ruohan Zhang, Weiyu Liu, Percy Liang, Li Fei-Fei, Jiayuan Mao, and Jiajun Wu. Embodied agent interface: Benchmarking llms for embodied decision making. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 100428–100534. Curran Associates, Inc., 2024a. doi: 10.52202/079017-3188. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b631da756d1573c24c9ba9c702fde5a9-Paper-Datasets_and_Benchmarks_Track.pdf.

Yijiang Li, Qingying Gao, Tianwei Zhao, Bingyang Wang, Haoran Sun, Haiyun Lyu, Robert D Hawkins, Nuno Vasconcelos, Tal Golan, Dezhi Luo, et al. Core knowledge deficits in multi-modal language models. *arXiv preprint arXiv:2410.10855*, 2024b.

Weichen Liu, Qiyao Xue, Haoming Wang, Xiangyu Yin, Boyuan Yang, and Wei Gao. Spatial reasoning in multimodal large language models: A survey of tasks, benchmarks and methods. *arXiv preprint arXiv:2511.15722*, 2025.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16488–16498, 2024.

OpenAI. Gpt-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026a.

OpenAI. Gpt-6 astra. https://openai.com/index/gpt-6-astra/, 2026b.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026a. URL https://qwen.ai/blog?id=qwen3.6-27b.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026c. URL https://qwen.ai/blog?id=qwen3.8.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind: Failing to translate detailed visual features into words. *arXiv preprint arXiv:2407.06581*, 2024.

Santhosh Kumar Ramakrishnan, Erik Wijmans, Philipp Kraehenbuehl, and Vladlen Koltun. Does spatial cognition emerge in frontier models? *arXiv preprint arXiv:2410.06468*, 2024.

Fan-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. Layoutvlm: Differentiable optimization of 3d layout via vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29469–29478, 2025a.

Peiwen Sun, Shiqiang Lang, Dongming Wu, Yi Ding, Kaituo Feng, Huadai Liu, Zhen Ye, Rui Liu, Yun-Hui Liu, Jianan Wang, et al. Spacevista: All-scale visual spatial reasoning from mm to km. *arXiv preprint arXiv:2510.09606*, 2025b.

Kimi Team and et al. Kimi k2.5: Visual agentic intelligence, 2026. URL https://arxiv.org/abs/2602.02276.

Haoming Wang, Qiyao Xue, and Wei Gao. Infinibench: Infinite benchmarking for visual spatial reasoning with customizable scene complexity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21594–21605, 2026.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *Advances in Neural Information Processing Systems*, 37:75392–75421, 2024.

Shengyuan Wang, Zhiheng Zheng, Yu Shang, Lixuan He, Yangcheng Yu, Fan Hangyu, Jie Feng, Qingmin Liao, and Yong Li. Urbanworld2.0: A multimodal agent framework for reality-aligned 3d world generation at city-scale. *arXiv preprint arXiv:2511.18005*, 2025a.

Weizhen Wang, Chenda Duan, Zhenghao Peng, Yuxin Liu, and Bolei Zhou. Embodied scene understanding for vision language models via metavqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22453–22464, 2025b.

Wenqi Wang, Reuben Tan, Pengyue Zhu, Jianwei Yang, Zhengyuan Yang, Lijuan Wang, Andrey Kolobov, Jianfeng Gao, and Boqing Gong. Site: towards spatial intelligence thorough evaluation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9058–9069, 2025c.

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

Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5922–5931, 2021.

Yuxi Xiao, Longfei Li, Shen Yan, Xinhang Liu, Sida Peng, Yunchao Wei, Xiaowei Zhou, and Bingyi Kang. Spatialtree : How spatial abilities branch out in MLLMs. In *The First Workshop on Efficient Spatial Reasoning*, 2026. URL https://openreview.net/forum?id=DUqZa55ElK.

Haotian Xu, Yue Hu, Zhengqiu Zhu, Chen Gao, Ziyou Wang, Junreng Rao, Wenhao Lu, Weishi Li, Quanjun Yin, and Yong Li. Citycube: Benchmarking cross-view spatial reasoning on vision-language models in urban environments. *arXiv preprint arXiv:2601.14339*, 2026.

Wenrui Xu, Dalin Lyu, Weihang Wang, Jie Feng, Chen Gao, and Yong Li. Defining and evaluating visual language models' basic spatial abilities: A perspective from psychometrics. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11571–11590, 2025.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10632–10643, 2025.

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

Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*, 2025.

Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Xiuyu Li, Michael J. Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning, 2026. URL https://arxiv.org/abs/2601.11109.

Songsong Yu, Yuxin Chen, Hao Ju, Lianjie Jia, Fuxi Zhang, Shaofei Huang, Yuhan Wu, Rundi Cui, Binghao Ran, Zaibin Zhang, et al. How far are vlms from visual spatial intelligence? a benchmark-driven perspective. *arXiv preprint arXiv:2509.18905*, 2025.

Jian Zhang, Shijie Zhou, Bangya Liu, Achuta Kadambi, and Zhiwen Fan. Spatialstack: Layered geometry-language fusion for 3d vlm spatial reasoning. *arXiv preprint arXiv:2603.27437*, 2026a.

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. Mapnav: A novel memory representation via annotated semantic maps for vlm-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13032–13056, 2025a.

Lingfeng Zhang, Yuening Wang, Hongjian Gu, Atia Hamidizadeh, Zhanguang Zhang, Yuecheng Liu, Yutong Wang, David Gamaliel Arcos Bravo, Junyi Dong, Shunbo Zhou, et al. Et-plan-bench: Embodied task-level planning benchmark towards spatial-temporal cognition with foundation models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 21566–21573. IEEE, 2025b.

Pingyue Zhang, Zihan Huang, Yue Wang, Jieyu Zhang, Letian Xue, Zihan Wang, Qineng Wang, Keshigeyan Chandrasegaran, Ruohan Zhang, Yejin Choi, et al. Theory of space: Can foundation models construct spatial beliefs through active exploration? In *The Fourteenth International Conference on Learning Representations*, 2026b.

Wenyu Zhang, Wei En Ng, Lixin Ma, Yuwen Wang, Junqi Zhao, Allison Koenecke, Boyang Li, and Lu Wang. SPHERE: Unveiling spatial blind spots in vision-language models through hierarchical evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11591–11609, Vienna, Austria, July 2025c. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.568. URL https://aclanthology.org/2025.acl-long.568/.

Yong Zhao, Kai Xu, Zhengqiu Zhu, Yue Hu, Zhiheng Zheng, Yingfeng Chen, Yatai Ji, Chen Gao, Yong Li, and Jincai Huang. Cityeqa: A hierarchical llm agent on embodied question answering benchmark in city space. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12476–12491, 2025.

Haoyu Zhen, Xiaolong Li, Yilin Zhao, Han Zhang, Sifei Liu, Kaichun Mo, Chuang Gan, and Subhashree Radhakrishnan. 3d-layout-r1: Structured reasoning for language-instructed spatial editing. *arXiv preprint arXiv:2603.22279*, 2026.

Weipeng Zhong, Peizhou Cao, Yichen Jin, Li Luo, Wenzhe Cai, Jingli Lin, Hanqing Wang, Zhaoyang Lyu, Tai Wang, Bo Dai, et al. Internscenes: A large-scale simulatable indoor scene dataset with realistic layouts. *arXiv preprint arXiv:2509.10813*, 2025.

A Appendix

A.1 Benchmark Design and Evaluation Details

A.1.1 Task Level Composition and Diagnostic Roles

SpatialAct consists of three complementary task levels designed to diagnose different stages of spatial intelligence. Rather than competing objectives, these levels provide a progressive evaluation hierarchy from static spatial reasoning to closed-loop interactive refinement. As Figure 2 shows, the benchmark consists of 2,600 Basic Spatial Ability questions, 1,200 Single-step Error Detection and Fix questions, and 555 Multi-turn Interactive Refinement cases. The Basic Spatial Ability tasks evaluate prerequisite spatial capabilities, such as spatial relations, mental rotation, and spatial visualization. These tasks isolate fundamental 3D understanding without requiring action planning or environment updates, allowing us to distinguish failures in basic perception and reasoning from failures caused by interaction complexity. The Single-step Error Detection and Fix task further tests whether these foundational spatial abilities are sufficient for selecting an executable correction without the confound of accumulated state changes. Finally, Multi-turn Interactive Refinement serves as the central integrative task, requiring models to repeatedly diagnose spatial errors, generate corrective actions, interpret feedback, and maintain a coherent spatial state throughout interaction.

A.1.2 Scene Distribution and Complexity

SpatialAct comprises 333 scene configurations, of which 97 are indoor and 236 are urban architectural. For the Multi-turn Interactive Refinement task, we construct 555 evaluation cases by injecting different combinations of spatial errors into selected scenes from this collection, yielding 161 indoor cases and 394 urban architectural cases. Abstract scenes are primarily designed to isolate elementary spatial abilities using simple primitives and controlled spatial relations. Their limited structural complexity and lack of contextual layout constraints make them less representative of practical iterative 3D scene refinement. We therefore use Abstract Geometric scenes in the basic diagnostic layer rather than in the primary interactive refinement task. The benchmark includes seven initially error-free cases, while the remaining scenes are grouped into three complexity levels with 1–3, 4–6, and 7+ initial errors, corresponding to 255, 215, and 78 cases, respectively.

A.1.3 Choice of Multi-turn Interactive Refinement

We choose Multi-turn Interactive Refinement to examine whether spatial competence can support sustained improvement when a model's own actions change the scene that subsequent decisions depend on. This task is not simply a longer sequence of independent spatial questions. Each edit can resolve existing errors, alter surrounding spatial relations, or introduce new violations, so a locally appropriate correction does not necessarily improve the scene as a whole. In the standard setting, models receive an erroneous scene and predefined validity criteria, rather than a target layout or a prescribed repair sequence. They must identify the remaining inconsistencies, determine what to modify, and reassess the scene after execution. Because success depends on satisfying geometric constraints rather than reproducing a unique reference configuration, different valid layouts and repair trajectories are accepted. The task therefore combines open-ended problem diagnosis and correction with objectively verifiable outcomes. Importantly, multi-turn feedback provides opportunities to recover from imperfect edits rather than treating the first unsuccessful action as the final outcome. The purpose is not to reward longer trajectories, but to test whether models can use feedback to make cumulative progress toward a valid scene. Working with existing scene assets and high-level transformations also keeps the evaluation focused on spatial refinement rather than asset synthesis or low-level motor control. Comparing performance across foundational tasks, the single-step task, and multi-turn refinement then provides a basis for investigating why model differences become more pronounced during interaction and which interventions can narrow the gaps between weaker agents, stronger models, and human reference performance.

A.1.4 Choice of Actions and Error Types

`Move`, `Rotate`, and `Scale` form a compact set of geometric transformations that modify object position, orientation, and size. These operations are directly executable by the simulator while

abstracting away low-level motor control. At the same time, they do not prescribe a specific repair strategy, as models must still determine the target object, operation type, direction or rotation sign, and continuous magnitude. Different errors may be resolved by different operations, and a single action may introduce new violations; therefore, SpatialAct does not enforce a fixed mapping between error types and actions.

Our error categories are designed around common geometric validity failures in 3D layouts. InfiniBench (Wang et al., 2026) identifies object overlap, out-of-boundary placement, and unrealistic orientation as three representative failure patterns in generated layouts, which correspond to the core geometric constraints evaluated in SpatialAct. Related layout generation and editing works, including Holodeck (Yang et al., 2024), LayoutVLM (Sun et al., 2025a), and FirePlace (Huang et al., 2025a), similarly consider geometric constraints such as collision avoidance, scene boundaries, wall relations, and alignment. Based on these common geometric constraints, SpatialAct defines Collision as object–object conflict, while Wall Conflict and Road Conflict represent domain-specific object–environment violations in indoor and urban scenes, respectively. Orientation Error captures violations of expected spatial alignment.

SpatialAct prioritizes objectively verifiable geometric errors, enabling consistent evaluation of how models diagnose and repair existing layouts through executable actions and iterative feedback. Semantic-level issues, such as whether an arrangement matches functional requirements or human preferences, represent a higher-level and more challenging aspect of spatial intelligence. We acknowledge their importance and consider them valuable directions for future investigation, but they are beyond the current scope of this benchmark.

A.1.5 Spatial Error Definitions and Evaluation Criteria

SpatialAct evaluates four types of spatial errors using deterministic geometric predicates with predefined eligibility filters and category-specific rules: Collision, Wall Conflict, Road Conflict, and Orientation Error. Collision represents object–object geometric violations, where two editable objects or buildings exhibit footprint or volume intersection beyond the specified criteria. Certain indoor object-category pairs are excluded through predefined whitelists to avoid counting semantically valid contacts as errors. Wall Conflict captures indoor object–environment violations caused by boundary penetration or excessive wall intersection. Road Conflict represents the corresponding urban constraint, where a building footprint intersects a road region. Orientation Error measures whether an eligible object's dominant axis deviates from its prescribed spatial alignment.

For Urban Architectural scenes, Collision is detected when the building footprint intersection exceeds $0.02\,\mathrm{m}^2$, while Road Conflict is detected when building–road footprint intersection exceeds $0.25\,\mathrm{m}^2$. Urban Orientation Error is triggered when the dominant axis deviates by at least 15° from both the estimated road direction and its perpendicular direction, subject to building-shape eligibility filters. For Indoor scenes, Collision requires an XY-plane intersection area of at least $10^{-4}\,\mathrm{m}^2$ together with at least five BVH overlap hits. Wall Conflict is triggered by category-aware boundary violations or when the wall-normal penetration-depth-to-thickness ratio exceeds 0.12. Indoor Orientation Error applies to eligible near-wall objects whose dominant axis deviates by more than 12° from the nearest global X/Y axis. The same error predicates and evaluation configuration are consistently applied to the initial scene, intermediate states after each executed action, and final scene states, ensuring comparable measurement of refinement progress throughout the interaction.

A.1.6 Basic Spatial Ability and Single-step Error Detection Subcategories

To provide a fine-grained diagnosis of model performance, we further divide the Basic Spatial Ability tasks and the Single-step Error Detection and Fix task into multiple subcategories. As shown in Table 3, the five Basic Spatial Ability tasks cover Object Meaning, Spatial Relation, Spatial Orientation, Mental Rotation, and Spatial Visualization, with each category containing several subtypes that emphasize different aspects of 3D spatial understanding and reasoning. In addition, the Single-step Error Detection and Fix task is divided according to different error patterns, enabling targeted evaluation of whether models can identify abnormal spatial configurations and select the corresponding correction. Representative prompts for each subcategory are provided in the Table 3.

Table 3: Subcategories and representative prompts for Basic Spatial Ability tasks and Single-step Error Detection and Fix tasks.

Task	Prompt
Object Meaning (7)	How many bins are in the room? What is object 1? Which numbered object is a commode? Which side of the box is the frame on? Which numbered object is east of the commode? Which numbered object is closest to the cabinet? Which numbered object is farthest from the cabinet?
Spatial Relation (4)	Which side is building 1 relative to building 5? Which two buildings are closest to each other? Which building is farthest from building 2? How many buildings are on the east side of building 3?
Spatial Orientation (3)	Consider the southernmost building in the isometric image. Relative to the scene center, which direction is that building located in the top-view image? An isometric image where only one building is labeled 'A', and a top-view image with numbered buildings. Which numbered building is 'A' in the top-view image? Identify the westernmost building in the image. Now imagine that the camera rotates CLOCKWISE by 90 degrees around the scene center.When the camera rotates, the cardinal directions (north, east, south, west) rotate together with the view. After this rotation, where will that building be located relative to the scene center in the rotated coordinate frame?
Mental Rotation (4)	If building 1 is rotated counterclockwise by 30 degrees around its own center, will it collide with building 4? Rotate building 1 counterclockwise by one of the angles below around its own center. Which rotation angle avoids collision with building 4? Building 1 is rotated clockwise by 300 degrees.Which image matches this rotation? In the original image, Building 6 is rotated clockwise by 270 degrees around the center of the region. Which option shows the correct result after this rotation?
Spatial Visualization (2)	If building 1 is moved West by 6 meters, will it collide with building 6? If we swap the positions of building 4 and building 5, will there be any collision in the scene AFTER the swap?
Single-step Error Detection and Fix (2)	Examine the scene carefully and identify what problem exists. Known issue: Building 1 has an angle anomaly near a road intersubsection. Choose ONE action to fix it in ONE step. The fix should resolve the issue without introducing new problems.

A.1.7 Multi-turn Interactive Refinement Prompt Design

The Multi-turn Interactive Refinement task evaluates closed-loop spatial reasoning and action over evolving scene states. At each interaction round, the model receives the current scene observations and the available interaction history, and is required to reason about the current spatial configuration and generate executable editing commands. After the actions are applied by the simulator, updated scene observations are returned to the model for the next round. This process continues until the model determines that no further correction is required or reaches the maximum interaction limit.

The prompt instructs the model to identify remaining spatial issues, determine appropriate corrective operations, and express its actions using the predefined executable format. The action space is shared across evaluated models and contains object movement, rotation, and scaling operations. A representative Multi-turn Interactive Refinement prompt and response format for indoor scenes are provided in Figure 6.

```
# Task Description

You are now in a follow-up round for indoor geometry correction.

Previous operations have already been applied, so compare original-vs-current
image pairs and identify remaining clear issues.

# Image Reference

- Top-view pair: Image 1 (ORIGINAL top-down) vs Image 3 (CURRENT top-down after previous
    actions)

- Isometric pair: Image 2 (ORIGINAL isometric) vs Image 4 (CURRENT isometric after previous
     actions)

# Wall Rendering Hint

- In top-view image: the white outer boundary represents walls.

- In isometric image: the translucent/transparent outer shell represents walls.

# Coordinate Reference

- Distance: use unit, with the visible white 1unit scale bar in the image as reference.

- Direction: use North / South / East / West according to the red N arrow in each image.

# Error Types (Indoor Only)

Check independently for each labeled object:

1. Overlap: physical intersection between object volumes/footprints.

2. WallConflict: object collides with room walls (intersects/penetrates wall boundary),
or extends outside wall bounds.

3. Orientation: for rectangular/non-round furniture, major axis is clearly misaligned
with room axis (not near 0/90 degrees).

# Your Analysis Steps

1. Compare each original/current image pair by view type.

2. For every labeled object, check all three error types independently.

3. For each previously detected and attempted error type:

   - FIXED: this specific error type is now clearly resolved.

   - STILL_HAS_ISSUE: this specific error type still exists or needs further adjustment.

   - WORSE: this specific error type became worse after previous action.

4. Also report newly visible issue types that were missed before.
For newly discovered error types, set issue_status to NEW.

# Important Rules

- Detect these three error_type values:
"Overlap", "WallConflict", "Orientation".

- One object may have multiple error types; output one entry per object per error type.

- You may propose actions ONLY for labeled IDs in the allowed set.

- Error validity is global: judge geometry on the WHOLE scene.

- Never resolve a labeled-object issue by creating new overlap/wall/orientation errors
on unlabeled objects.

# Output Requirements

- Return ONLY a JSON array.

- One object = one object id + one error type.

Each item:
```

```
- id: integer object label

- issue_status:
"FIXED" | "STILL_HAS_ISSUE" | "WORSE" | "NEW"

- error_type:
"Overlap" | "WallConflict" | "Orientation"

- reason: concise status explanation for this error type

- action: incremental fix actions from current state (Move/Rotate/Scale)

# Action Command Semantics

- Move(direction, distance): move in the current image compass frame.

- Rotate(angle): rotate around object center on Z axis.

- Scale(factor): uniform XYZ scaling around object center.

# Previous Iteration Actions (JUST APPLIED)

- Building 1 (Orientation): Rotate(10)

- Building 2 (Orientation): Rotate(8)

- Building 3 (Orientation): Rotate(12)

- Building 6 (Orientation): Rotate(20)

- Building 7 (Overlap): Move(West, 0.2)

- Building 8 (WallConflict): Move(South, 0.3)

- Building 8 (Orientation): Rotate(-33)

# Cumulative Adjustments Summary

- id=1: Rotate(10)

- id=2: Rotate(8)

- id=3: Rotate(12)

- id=6: Rotate(20)

- id=7: Move(West, 0.2)

- id=8: Move(South, 0.3), Rotate(-33)

Only use ids from this allowed set:
[1, 2, 3, 4, 5, 6, 7, 8]

Labeled objects:
object 1 (couch), object 2 (chair), object 3 (chair),
object 4 (lamp), object 5 (wardrobe),
object 6 (chair), object 7 (lamp), object 8 (chair)

# Example Response

[
  {
    "id": 1,
    "issue_status": "FIXED",
    "error_type": "Orientation",
    "reason": "The couch's major axis is now nearly aligned with the room axis.",
    "action": []
  },
  {
    "id": 8,
    "issue_status": "STILL_HAS_ISSUE",
    "error_type": "Orientation",
    "reason": "Chair 8 remains approximately 24 degrees clockwise from alignment.",
    "action": ["Rotate(-24)"]
  }
]
```

Figure 6: Example Multi-turn Interactive Refinement prompt and response format for indoor scenes.

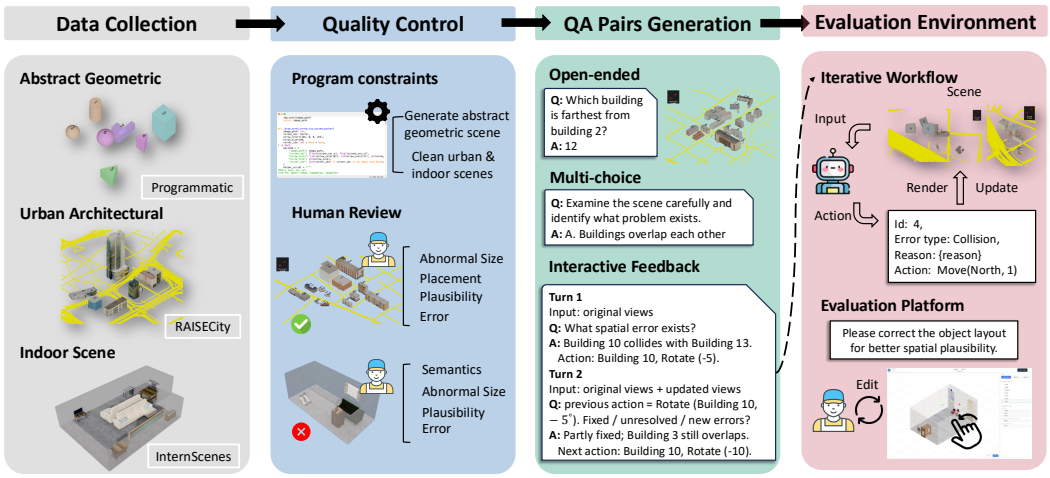


Figure 7: Benchmark construction pipeline, including data collection, quality control, QA pair generation, and evaluation environment setup.

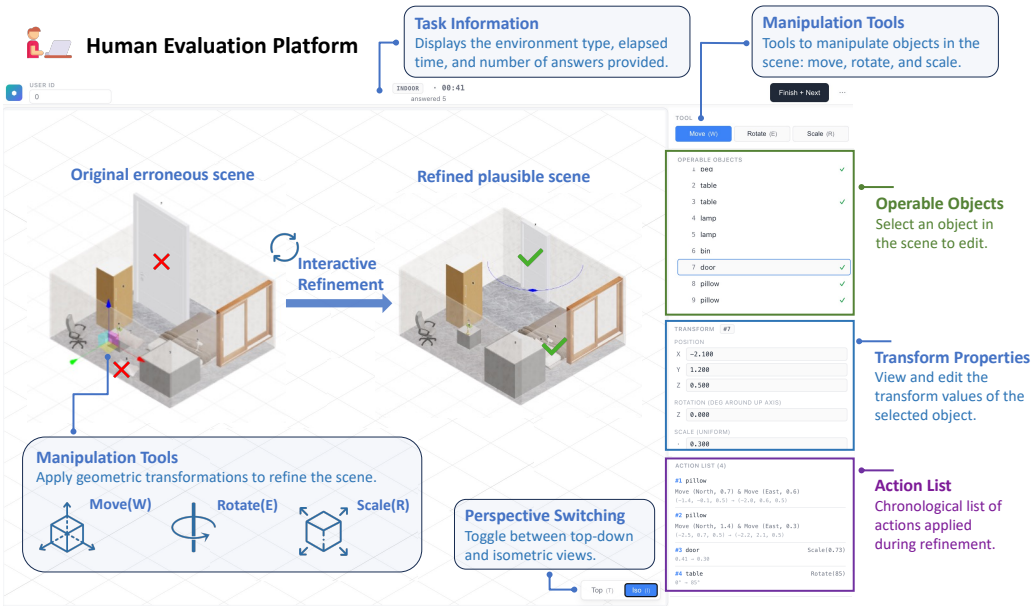


Figure 8: The web-based evaluation interface for interactive scene refinement.

A.2 Benchmark Construction Details

SpatialAct is constructed through a multi-stage pipeline involving scene collection, quality control, QA generation, and evaluation environment preparation. As illustrated in Figure 7, we first collect 3D scenes from procedural generation and existing scene resources, then perform programmatic and manual quality checks to ensure valid spatial layouts. Based on the processed scenes, we generate QA pairs for different task levels and configure the simulator-based interaction environment for multi-turn refinement.

A.3 Human Evaluation Platform and Protocol

To support human evaluation on the Multi-turn Interactive Refinement task, we develop a web-based 3D editing interface, as illustrated in Figure 8. The platform is designed to provide an interactive workflow for human participants to inspect and correct spatial inconsistencies in generated scenes. Participants observe each scene through the same fixed top-down and isometric views provided to the evaluated models, ensuring consistent visual evidence during refinement. The interface provides a labeled Object Checklist for selecting scene entities and allows participants to modify object properties, including position, rotation, and scale. Selected objects can be adjusted either through

an interactive Transform Gizmo or through the numerical Transform Properties panel for precise manipulation. In addition, an Operation History module records previous modifications, enabling participants to track their refinement process during multi-turn interaction.

The human evaluation follows the same refinement objective and transformation space as the model setting. Specifically, both human participants and VLM agents receive identical scene observations, object labels, and available operations, including object movement, rotation, and scaling. The main difference lies in the action execution interface: human participants can directly manipulate objects through the graphical interface, whereas models generate transformations using the predefined textual action grammar. This difference arises from the current limitation that VLM agents cannot directly operate continuous 3D manipulation interfaces. To bridge this interaction gap, we further introduce Simulator-guided Lookahead for World Model Verification in section 3.4, which augments the agent with a trial-and-refinement mechanism analogous to human editing workflows by allowing tentative edits to be simulated and evaluated before execution. Therefore, we treat human performance as a feasibility reference for interactive scene refinement rather than a strictly action-interface-matched comparison. Seven participants independently performed the Multi-turn Interactive Refinement task using the proposed interface. Across all participants, the numbers of completed refinement trajectories were 61, 61, 31, 62, 301, 304, and 61, respectively, resulting in 881 valid submissions over 555 benchmark scenes. For scenes completed by multiple participants, we first aggregate the results within each scene and then average across all benchmark scenes, ensuring that every scene contributes equally to the final human performance measurement.

A.4 Metric Design in Main Results

For the Basic Spatial Abilities tasks and the Single-step Error Detection and Fix task, we use accuracy as the evaluation metric, which measures whether the model can correctly understand spatial concepts or identify appropriate scene modifications.

For the Multi-turn Interactive Refinement task, we design five metrics to evaluate two complementary aspects of model performance: repair effectiveness and interaction efficiency. For repair effectiveness, we first define Repair Rate to measure the model's ability to reduce scene errors:

$$\text{Repair Rate} = \frac{E_{\text{before}} - E_{\text{after}}}{E_{\text{before}}}, \tag{1}$$

where E_{before} and E_{after} denote the number of errors in the scenes before and after model refinement, respectively. This metric directly measures the relative reduction of spatial errors after interaction.

Second, we define Scene Success Rate to evaluate whether the model can completely resolve all errors at the scene level:

$$\text{Scene Success Rate} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathbb{I}(E^s_{\text{after}} = 0), \tag{2}$$

where E^s_{after} denotes the number of remaining errors in scene s after refinement. This metric evaluates the proportion of scenes that are fully repaired.

To further measure the final refinement quality beyond discrete error counts, we introduce Scene MAU, which evaluates the normalized improvement between the initial and final scene states. The detailed definition of Scene MAU and its relationship with atomic- and round-level MAU are provided in Appendix A.7.

For interaction efficiency, we define Effective Repair Turn Ratio to measure the proportion of refinement turns that successfully reduce scene errors:

$$\text{Effective Repair Turn Ratio} = \frac{\sum_{s \in \mathcal{S}} \sum_{t=1}^{T_s} \mathbb{I}(E^s_t < E^s_{t-1})}{\sum_{s \in \mathcal{S}} T_s}, \tag{3}$$

where E^s_t denotes the number of remaining errors after the t-th refinement turn, and T_s denotes the total number of interaction turns for scene s. This metric reflects how frequently the model produces effective actions during refinement.

Finally, we report Average Completion Tokens per Scene to measure the reasoning cost:

$$\text{Avg. Completion Tokens} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} C_s, \tag{4}$$

where C_s denotes the total number of completion tokens generated by the model across all interaction turns for scene s. A lower value indicates that the model achieves refinement with fewer generated tokens.

A.5 Domain-Specific Spatial VLM Baselines on Multi-turn Refinement

To further evaluate whether models specialized for spatial understanding can better support interactive scene refinement, we include three domain-specific spatial VLM baselines: Qwen3-VL-2B-XVR (Jeong et al., 2026), SenseNova-SI-1.3-Qwen3-VL-8B (Cai et al., 2026), and SpatialStack-Qwen3.5-4B (Zhang et al., 2026a). These models are evaluated under the same Multi-turn Interactive Refinement protocol as the general-purpose VLM baselines.

Table 4 reports their performance across Repair Rate, Scene Success Rate, Effective Repair Turn Ratio, severity-aware Scene MAU, and Average Completion Tokens. Despite their specialized 3D representations or spatial modeling capabilities, these models do not consistently outperform general-purpose VLMs in the interactive refinement setting. All three baselines achieve negative Repair Rates and negative Scene MAU values, indicating that their generated actions often fail to reduce existing geometric errors and may introduce additional violations during iterative refinement. These results suggest that specialized 3D spatial modeling capabilities alone do not necessarily lead to effective closed-loop refinement, which requires maintaining scene states and adapting actions based on interaction feedback.

Table 4: Performance of domain-specific spatial VLM baselines on Multi-turn Interactive Refinement.

Model	Rep. Rate ↑	Succ. Rate ↑	Eff. Turn ↑	Scene MAU ↑	Tok./ Scene
Qwen3-VL-2B-XVR	-1.247	0.004	0.262	-0.482	11903
SenseNova-SI-1.3-Qwen3-VL-8B	-0.848	0.002	0.167	-0.414	10531
SpatialStack-Qwen3.5-4B	-0.319	0.009	0.151	-0.182	4063

A.6 Diagnostic Intervention Details

To diagnose the sources of failure in Multi-turn Interactive Refinement, we design three controlled interventions and one auxiliary analysis based on Gemini-3.1 Pro Preview. All variants use the same iterative repair pipeline with a maximum of 30 interaction rounds, and final performance is evaluated by applying the original geometry detector to the initial and final scenes using the complete object-label set.

−**Perception.** This intervention removes predefined instance grounding and evaluates the model's ability to recover operable object identities from visual observations. Specifically, this setting is applied only to Indoor scenes, where semantic object categories and instance identities are available. The model first receives unnumbered top-down and isometric views and predicts the number of visible instances for each category from a candidate list containing all scene categories as well as four absent distractor categories. The predicted counts are then converted into a new set of object labels, which are used throughout the subsequent iterative repair process. Therefore, this intervention isolates the impact of object grounding and visual-semantic identification while keeping the remaining repair pipeline unchanged.

+**Reasoning.** This intervention provides oracle-level error diagnosis while preserving the model's action generation capability. At each iteration, a geometry detector supplies the current set of $(\text{object ID}, \text{error type})$ pairs, including overlap, wall conflict, and orientation errors. The model therefore bypasses the error identification step and focuses on determining appropriate repair actions from the scene observations. After each executed action, the detector is rerun so that newly introduced errors are incorporated into subsequent iterations.

−**Memory.** This intervention removes all explicit cross-round information and treats each interaction step as an independent request. At each iteration, the model receives only the current top-down and isometric views, without previous actions, cumulative adjustment summaries, original-to-current image comparisons, or prior model responses. The rendered scene itself still reflects all previously executed actions, ensuring that the intervention removes memory access rather than environmental state changes.

Invalid Action Analysis. We additionally analyze invalid action generation to determine whether refinement failures originate from producing unusable commands. This analysis follows the same execution pipeline and examines whether non-empty responses can be parsed into valid actions and executed by the simulator. We categorize abnormal responses into five types: (1) malformed JSON outputs, (2) JSON arrays containing non-object elements, (3) outputs missing the required `action` field, (4) outputs whose `action` field violates the predefined schema, and (5) action commands

Table 5: Invalid response analysis across evaluated models.

Model	Non-empty Resp.	Abnormal Resp.	Abn. Rate
Proprietary Models			
GLM-5V-Turbo	9,806	4	0.041%
GPT-5.4 mini	1,290	7	0.543%
GPT-5.4	2,753	26	0.944%
Gemini-3.1 Pro	8,784	5	0.057%
GPT-6 Astra	1,499	0	0.000%
Open-source Models			
DeepSeek-V4.1-Flash	2485	21	0.845%
Kimi-K2.5	110	10	9.091%
Qwen3.6-27B	46	4	8.696%
Qwen3.6-35B-A3B	3,756	146	3.887%
Qwen3.8-27B	1,832	1	0.055%
Domain-specific Spatial Models			
Qwen3-VL-2B-XVR	12,610	3	0.024%
SenseNova-SI-1.3-Qwen3-VL-8B	13,450	9	0.067%
SpatialStack-Qwen3.5-4B	7,824	702	8.972%

whose format does not match the simulator interface. Table 5 summarizes the abnormal response statistics across evaluated models. Most proprietary models exhibit extremely low abnormal response rates, while several open-source and domain-specific spatial models encounter more frequent output-format failures. However, these failures occur relatively infrequently compared with the overall interaction process and cannot fully account for the large performance differences observed in multi-turn refinement. These results suggest that invalid action generation alone cannot explain the gap between weaker and stronger models; instead, the remaining gap is likely attributed to higher-level capabilities required for reliable multi-turn refinement, including spatial reasoning, effective action execution, and persistent state maintenance.

A.7 Severity-Aware Evaluation with Marginal Action Utility

Repair Rate, Scene Success Rate, and Effective Repair Turn Ratio capture whether errors are reduced, eliminated, or repaired efficiently, but they do not quantify the magnitude of geometric improvement or deterioration. We therefore introduce Marginal Action Utility (MAU), a severity-aware metric that measures continuous changes in scene quality at three temporal resolutions: individual atomic actions, complete interaction rounds, and final scene outcomes.

Scene Severity Modeling Let x denote a scene state and let

$$\mathcal{T} = \{\text{Overlap}, \text{RoadConflict}, \text{WallConflict}, \text{Orientation}\}$$

be the set of evaluated error types. Each error instance e is assigned a continuous normalized severity $s_e(x) \in [0, 1]$. When comparing two scene states, an error instance that is absent in one state is assigned a severity of zero. All severity values are finally clipped to the range $[0, 1]$.
For overlap errors, severity is defined as the intersubsection area between two object footprints divided by the smaller footprint area. For road conflicts, severity is computed as the building penetration depth along the local road normal divided by the corresponding road width. When a building intersects multiple road components, the maximum component-wise ratio is used. For wall conflicts, severity is defined as the maximum boundary violation depth, including wall penetration and enclosure-boundary violation, normalized by the thickness of the nearest wall. A binary out-of-bound conflict without a reliable continuous depth is assigned severity one. For orientation errors, given the axial orientation error δ_e and detector tolerance τ, the severity is computed as

$$\frac{\delta_e - \tau}{45^\circ - \tau}.$$

The total geometric severity of a scene is defined as

$$G(x) = \sum_{t \in \mathcal{T}} G_t(x),$$

where

$$G_t(x) = \sum_{e:\mathrm{type}(e)=t} s_e(x).$$

All error instances within the predefined evaluation scope are re-detected after every action. Therefore, MAU measures actual geometric changes in the scene and does not depend on the error type predicted by the model or assume any predefined correspondence between actions and error categories.

Atomic MAU For an atomic action a_i, let x_i^- and x_i^+ denote the scene states immediately before and after execution. For all error instances observed in either state, we compute the severity change as

$$\Delta_{i,e} = s_e(x_i^-) - s_e(x_i^+).$$

The total geometric benefit and harm associated with the action are

$$B_i = \sum_e \max(\Delta_{i,e}, 0),$$

$$H_i = \sum_e \max(-\Delta_{i,e}, 0).$$

The resulting net geometric utility is

$$U_i = B_i - H_i = G(x_i^-) - G(x_i^+).$$

We define Atomic MAU as

$$\mathrm{AtomicMAU}_i = \frac{G(x_i^-) - G(x_i^+)}{\max\{G(x_i^-), G(x_i^+)\}}.$$

The score is bounded within $[-1, 1]$. A positive value indicates net reduction of geometric severity, zero indicates no net change, and a negative value indicates that the action introduces more geometric damage than it removes. This formulation provides partial credit for actions that improve an error without completely resolving it.

For model-level reporting, Mean Atomic MAU is computed over all executable atomic actions:

$$\overline{\mathrm{AtomicMAU}} = \frac{1}{|\mathcal{A}^+|} \sum_{i \in \mathcal{A}^+} \mathrm{AtomicMAU}_i \,.$$

Round MAU Atomic actions generated within the same interaction round form a complete repair plan. Let $x_{s,r}^-$ and $x_{s,r}^+$ denote the canonical scene states before and after all actions in round r of scene s, respectively. Round MAU is defined as

$$\mathrm{RoundMAU}_{s,r} = \frac{G(x_{s,r}^-) - G(x_{s,r}^+)}{\max\{G(x_{s,r}^-), G(x_{s,r}^+)\}}.$$

Mean Round MAU is calculated over all actionable rounds:

$$\overline{\mathrm{RoundMAU}} = \frac{1}{|\mathcal{R}^+|} \sum_{(s,r) \in \mathcal{R}^+} \mathrm{RoundMAU}_{s,r} \,.$$

Unlike Atomic MAU, Round MAU evaluates the net effect of a complete round-level repair plan and is independent of how many atomic actions are used to represent that plan.

Scene MAU To evaluate end-to-end refinement quality, we further define Scene MAU between the initial and final states of each scene. Let $x_{s,0}$ and $x_{s,F}$ denote the initial and final canonical states of scene s:

$$\mathrm{SceneMAU}_s = \frac{G(x_{s,0}) - G(x_{s,F})}{\max\{G(x_{s,0}), G(x_{s,F})\}}.$$

A value of one indicates complete removal of geometric severity, a positive value indicates partial improvement, zero indicates no net change, and a negative value indicates that the final scene becomes worse than the initial state. If a model takes no action in an eligible scene, then $x_{s,F} = x_{s,0}$ and the corresponding Scene MAU is zero.

Mean Scene MAU is computed by averaging over all eligible scenes:

$$\overline{\mathrm{SceneMAU}} = \frac{1}{|\mathcal{S}^+|} \sum_{s \in \mathcal{S}^+} \mathrm{SceneMAU}_s \,.$$

Atomic, Round, and Scene MAU share the same normalized scale but capture different aspects of refinement. Atomic MAU measures local action quality, Round MAU measures the quality of a complete repair step, and Scene MAU measures the final end-to-end refinement outcome.

Table 6: Severity-aware evaluation using Atomic, Round, and Scene MAU.

Model	Atomic Actions	Actionable Rounds	Mean Atomic MAU	Mean Round MAU	Mean Scene MAU
Proprietary Models					
GLM-5V-Turbo	21,169	9,408	0.0001	0.0022	0.0308
GPT-5.4 mini	1,278	777	0.0338	0.0531	0.0688
GPT-5.4	4,241	2,250	0.0381	0.0633	0.2079
Gemini-3.1 Pro	22,577	8,292	0.0130	0.0332	0.3607
GPT-6 Astra	3,212	1,055	**0.2156**	**0.4654**	**0.7554**
Open-source Models					
DeepSeek-V4.1-Flash	2,234	1,271	0.0277	0.0445	0.0997
Kimi-K2.5	238	91	0.0089	0.0199	0.0026
Qwen3.6-27B	111	41	0.0321	0.0578	0.0046
Qwen3.6-35B-A3B	10,328	3,151	-0.0031	-0.0075	-0.0530
Qwen3.8-27B	2,234	1,271	0.0277	0.0445	0.0997
Domain-specific Spatial Models					
Qwen3-VL-2B-XVR	70,532	12,159	-0.0062	-0.0237	-0.4822
SenseNova-SI-1.3-Qwen3-VL-8B	84,750	12,230	-0.0040	-0.0226	-0.4140
SpatialStack-Qwen3.5-4B	28,531	7,417	-0.0051	-0.0153	-0.1819

Case 1: Goal Achieved

```
## Current Goal Status
Goal: A scene with no geometric errors.
Evaluated state: The CURRENT scene, after applying your previous
round's actions.
Result: No geometric errors remain.
Goal status: ACHIEVED
Next step: Return [] to finish the task; no further repair
actions are needed.
```

Case 2: Goal Not Achieved

```
## Current Goal Status
Goal: A scene with no geometric errors.
Evaluated state: The CURRENT scene, after applying your previous
round's actions.
Result: Geometric errors remain. The repair is not complete.
Goal status: NOT ACHIEVED
Next step: Inspect the CURRENT images and propose actions to
repair the remaining errors.
```

Figure 9: Goal completion feedback inserted into the following-round prompt.

Results Table 6 reports the severity-aware evaluation results. GPT-6 Astra achieves the strongest performance across all three resolutions, obtaining the highest Atomic, Round, and Scene MAU. Its increasing utility from individual actions to complete scenes indicates that locally beneficial edits can accumulate into more effective repair trajectories. Gemini-3.1 Pro achieves the second-highest Scene MAU, although its Atomic MAU is lower than GPT-5.4 and GPT-5.4-mini. This difference highlights that local action quality and final scene-level effectiveness capture complementary properties of refinement. These results demonstrate that severity-aware MAU provides additional information by capturing the magnitude of geometric changes throughout interaction.

A.8 Goal Completion Feedback Formulation

Goal completion feedback is introduced at the output side of each interaction round. After the current-round actions are evaluated, the feedback is inserted into the prompt of the following round. It only reports whether geometric errors remain and provides no information about their locations, types, or corrective actions. The exact feedback formulation is shown in Figure 9.

A.9 Capability Dynamics: Metrics and Fixed-Cohort Analysis

To analyze how different capabilities evolve during multi-turn refinement, we define three complementary metrics covering current-scene reasoning, cumulative repair effectiveness, and spatial state tracking. These metrics are evaluated across interaction rounds under a matched-scene protocol. At interaction round t, we compare two models only on shared scene instances whose runs under both models have reached round t. Examples from the three scene settings are pooled together. For each metric, we average scene-level scores over scene pairs with valid measurements for both models.

Reasoning The reasoning metric measures how accurately a model identifies active spatial errors in the current scene before executing the actions of the corresponding round. Let $P_{s,t}$ and $E_{s,t}$ denote the predicted and simulator-detected active error sets for scene s at round t, respectively. We compute the error-set similarity using Jaccard similarity:

$$R_{s,t} = \frac{|P_{s,t} \cap E_{s,t}|}{|P_{s,t} \cup E_{s,t}|}.$$

Each error is represented as an object–error-type unit, except for overlap errors, which are represented as unordered object pairs and counted once. When both the predicted and ground-truth error sets are empty, the score is defined as one. Higher values indicate more accurate recognition of the current scene errors.

Action The action metric measures cumulative repair effectiveness after executing actions up to round t. Let $N_{s,0}$ denote the number of initial errors in scene s, and let $N_{s,t}$ denote the remaining error count after completing round t:

$$A_{s,t} = \frac{N_{s,0} - N_{s,t}}{N_{s,0}}.$$

Positive values indicate net error reduction, zero indicates no change, and negative values indicate that the interaction has introduced more errors than it removed. Scores are reported as percentages.

Spatial State The spatial state metric measures whether the model maintains and utilizes a coherent representation of previously observed object states, error status, action history, and repair outcomes. Starting from round 2, GPT-5.6-sol is used as an external judge to assign a holistic score between 0 and 100 based on the model response at the target round. The judge evaluates four aspects: tracking unresolved issues, maintaining object–error associations, remaining consistent with previous actions, and adapting decisions based on prior attempts. Scores range from severe state tracking failures (0–24) to accurate and effective state utilization without material lapses (90–100).

Fixed-Cohort Capability Dynamics The main analysis of capability dynamics is based on matched scenes that remain available at each interaction round, which may introduce changes in the evaluated cohort as some trajectories terminate early. To verify that the observed trends are not solely caused by scene attrition, we perform a fixed-cohort analysis by restricting each model to trajectories that reach all 30 interaction rounds. We construct model-specific fixed cohorts containing 139 scenes for Gemini-3.1 Pro and 221 scenes for GLM-5V-Turbo. For each model, the same metric definitions and evaluation procedures as the main analysis are applied using a time-invariant set of valid scenes. As shown in Figure 10, the overall dynamics remain consistent with the main analysis. Gemini-3.1 Pro exhibits a gradual decline in error recognition, with the error-set Jaccard decreasing from 0.379 at the first round to 0.282 at the final round, while GLM-5V-Turbo remains lower with moderate fluctuations. In terms of repair progress, Gemini-3.1 Pro recovers from early deterioration and reaches a 12.5% cumulative error reduction by round 30, whereas GLM-5V-Turbo remains negative throughout the interaction and ends at -32.9%. Spatial-state scores for both models decrease during early rounds and subsequently fluctuate, ending at 83.6 and 75.3, respectively. These fixed-cohort results preserve the key observations from the main analysis, showing that the degradation or instability of reasoning and state tracking is not solely explained by changing scene populations across rounds. Since the two fixed cohorts are model-specific and not restricted to their intersubsection, the results are intended to validate within-model temporal dynamics rather than provide a matched comparison between models.

A.10 Detailed Memory Design Settings

We study how different memory designs support persistent spatial state maintenance during iterative refinement. All memory variants use the same repair task, action space, and stopping criterion, and

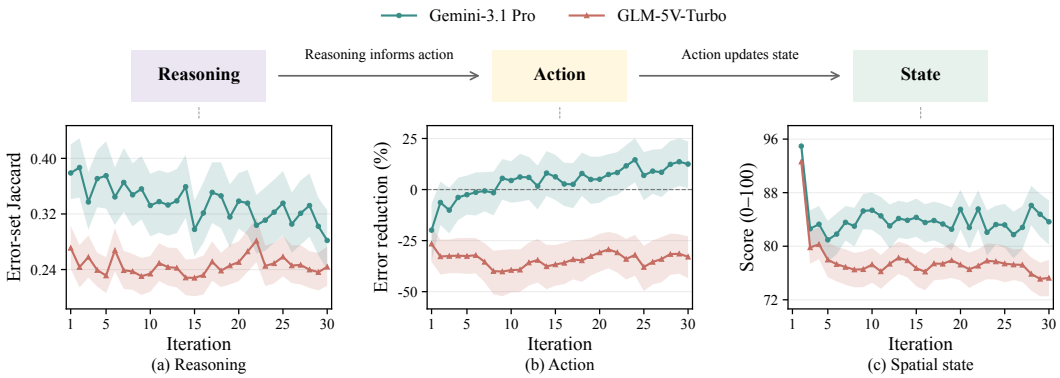


Figure 10: Capability dynamics on model-specific fixed 30-round cohorts. The analysis restricts each model to trajectories reaching all 30 rounds, with shaded regions indicating pointwise 95% scene-bootstrap confidence intervals computed separately for each model.

Table 7: Summary of memory designs.

Setting	Source	Memory Representation	Later-round Context	Output
−Memory	None	None	Current views only	Actions
M0	System	Action history + adjustments	Views + action summaries	Actions
M1	External	M0 + temporal 2D map	Views + summaries + map	Actions
M2	External	Metric 3D object state	Views + 3D state + actions	Actions
M3	Model	Qualitative cognitive map	Views + previous map + actions	Map + actions
M4	Model	Action–outcome lessons	Views + actions + lessons	Lessons + actions

differ only in the information maintained across interaction rounds and, when applicable, the response format used to update the memory. Memory is maintained independently for each scene and is never transferred across scenes. Table 7 summarizes the evaluated memory variants. The −Memory setting follows the same memoryless configuration used in the diagnostic intervention and removes all cross-round information, serving as a strict memoryless control. M0 corresponds to the default benchmark setting, where the system maintains previous actions and cumulative adjustments. M1 and M2 are externally maintained memories constructed from simulator states, while M3 and M4 require the model to maintain its own qualitative spatial beliefs or action–outcome experiences.

Common Protocol We evaluate all memory variants on the same set of 15 mixed-domain cases. The cases cover different difficulty levels. All experiments use Gemini-3.1 Pro Preview, with a maximum of 30 refinement rounds. The available actions are $\mathrm{Move}(\mathrm{direction}, \mathrm{distance})$, $\mathrm{Rotate}(\mathrm{angle})$, and $\mathrm{Scale}(\mathrm{factor})$. A run terminates when the model returns no actionable repair or reaches the maximum round limit. Except for the strict −Memory control, the first round starts from the initial top-down and isometric views. In subsequent rounds, the default setting receives the original and current views together with previously recorded information. Memory-specific text or structured representations are appended before the output requirements, while the action format and valid object IDs remain unchanged across settings.

−Memory The strict memoryless control follows the same −Memory configuration described in Appendix A.6. It removes all cross-round information. Each round is treated as an independent request containing only the current top-down and isometric views. Previous actions, cumulative adjustment summaries, original–current view comparisons, and explicit memory states are removed. The rendered scene still reflects all previously executed actions, ensuring that only access to historical information is removed.

M0: Benchmark Action Memory M0 corresponds to the default SpatialAct multi-turn setting. From the second round onward, the prompt includes two system-generated summaries:
(1) Previous Iteration Actions, which records the object IDs, associated error types, and commands executed in the immediately preceding round.

(2) Cumulative Adjustments Summary, which records consolidated transformations applied to each active object across completed rounds.
These summaries describe executed operations rather than inferred outcomes. The model must still verify their effects from the current scene views and outputs only the next action array.

M1: Temporal 2D Map Memory M1 (Zhang et al., 2025a) extends M0 with an externally constructed temporal top-down map. The map is generated from Blender geometry in a fixed scene coordinate frame. Each labeled object is represented by a simplified oriented footprint with a stable color and corresponding object ID. The current state is represented by solid outlines, while the state before the latest completed round is represented by dashed outlines. Arrows indicate measured planar translations across rounds, with older trajectories displayed with lower visual emphasis. The map is updated after each completed round. It may expose object identities that are visually occluded, but the model is instructed to verify physical constraints using scene images because simplified footprint overlap alone does not guarantee a valid collision. The model does not modify the map; the output remains the standard action format. An example of the temporal top-down map is shown in Figure 11.

M2: Metric 3D State Memory M2 (Fan et al., 2025) provides an externally measured metric representation generated from Blender after each completed round. All evaluated mesh parts belonging to the same object are merged into a single object record. The state contains the number of labeled objects and their geometric properties, including object ID, bounding box, center position, size, and horizontal orientation. Coordinates use a fixed scene frame with X pointing East, Y pointing North, and Z pointing upward. Distances follow the same units as the Move operation. Bounding boxes are computed over all evaluated vertices after transformations. The first round receives the initial measured state, while later rounds receive the updated state together with the previous-round actions. The model only consumes the metric state and does not modify it.

M3: Self-maintained Cognitive Map M3 (Zhang et al., 2026b; Ju et al., 2026) evaluates whether the model can construct and update its own qualitative representation of the scene. No memory is provided in the first round. Instead, the model infers a cognitive map from the top-down and isometric views. The cognitive map contains object identities, qualitative positions, qualitative orientations, and spatial notes describing important relations such as intersubsections, boundary constraints, alignments, or uncertainty. The model is encouraged to avoid unsupported precise coordinates or angles and may use unknown values when visual evidence is insufficient. In later rounds, the previous cognitive map and previous actions are provided. Since the map was generated before the previous action execution, the model must compare it against the current views, revise inconsistent beliefs, and output the updated map before generating new actions. The system stores the returned structure but does not modify or summarize it.

M4: Self-maintained Experience Memory M4 (Li et al., 2026) stores model-generated lessons extracted from previous action–outcome pairs. Unlike external memories, the system does not generate or revise these lessons. The first round is identical to M0 without any experience memory. Starting from the second round, the model compares previous actions and observed outcomes with the current scene and generates lessons before selecting new actions. In later rounds, previously stored lessons are provided and can be retained, revised, removed, or extended based on newly observed outcomes. Each lesson should describe an action–outcome relationship and its applicable conditions. The model is instructed to preserve uncertainty when causal attribution is unclear and avoid unsupported generalization. At most six concise lessons are maintained.
M1 and M2 represent external memory, where the system reconstructs memory states from simulator information and the model only consumes them. M3 and M4 represent self-maintained memory, where the model writes the memory returned in future rounds. M3 stores beliefs about the current spatial configuration, whereas M4 stores reusable experience about previous interventions.

A.11 Case Study and Error Analysis

To illustrate the challenges faced by current VLMs in SpatialAct, we present two representative error cases of GLM-5V-Turbo in Figure 12. These cases highlight how the model fail not only in spatial understanding and reasoning but also in subsequent action command generation. In the first case, set in an indoor scene, the model exhibits clear perceptual and reasoning errors. For instance, it incorrectly judges the door to be aligned within the wall, despite clearly protruding beyond the bounds. Moreover, the model erroneously infers that a curtain is soft and therefore cannot cause a conflict, failing to leverage the visual information available in the scene. In the second case, set in an

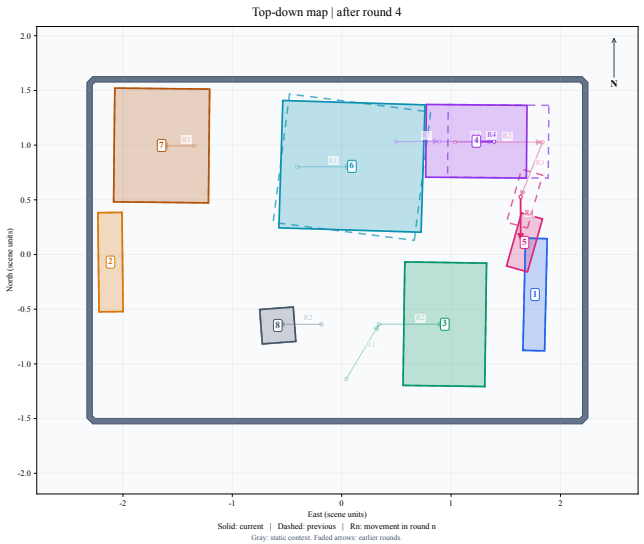


Figure 11: Example of the temporal top-down map used in M1. Solid outlines represent current object states, dashed outlines indicate states before the latest completed round, and arrows visualize planar translations across rounds.

GLM-5V-Turbo



Erroneous understanding and reasoning

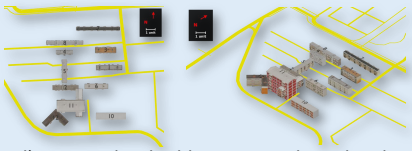
Reasoning:

…… Object 2 (stool): Circular object in the center-bottom area. Looks properly placed. Object 3 (curtain): The wavy/curved structure on the right side. This appears to be a curtain or fabric element. It seems to be within bounds. Object 4 (door): Located in the upper area, appears to be a door on the north wall. This looks like it's part of the wall structure. Object 5 (toilet): Located on the far right edge of the image. Looking at it carefully - this appears to be right at or possibly extending beyond the wall boundary on the east side. …… The toilet (object 5) sits very close to the eastern boundary and may actually extend past the wall line. The shelf (object 8) also appears positioned near that same edge. In the isometric view, both objects seem to be at or beyond where the transparent wall surface ends, suggesting they might have wall conflict issues. The door (object 4) appears integrated into the north wall as expected, so that placement looks correct. The curtain (object 3) is a soft, flexible element that doesn't create hard geometry conflicts. For the rectangular objects like the shelf, I need to verify whether their orientation aligns properly with the room's axes.

Actions:

Object 3: Move(West, 0.3)

Building 8: Move (Southwest, 0.4)



Erroneous action

Reasoning:

…… Building 12 appears to be rotated at an angle (not aligned with grid). Most other buildings seem aligned with the grid. Building 5: This vertical building appears to have a yellow road running through its lower portion. This is a RoadViolation. Building 5 sits directly on top of a horizontal yellow road, which is a clear violation. Building 12 is rotated approximately 45 degrees relative to the road grid, making it misaligned with the surrounding orientation.

Actions:

Building 5: Move(North, 0.8)

Building 12: Rotate (-45)

Figure 12: Representative error cases of GLM-5V-Turbo in SpatialAct.

urban architectural scene, the model correctly perceives and reasons that Building 5 intersects with a vertical road, identifying a spatial conflict. However, it subsequently executes an incorrect action, demonstrating a disconnect between reasoning and action.

A.12 Influence of Context and Task Relationship on VLM Behavior

Effect of Context Window Size As illustrated in Figure 13a, we analyze context-window effects on Kimi-K2.5, a model that tends to produce long reasoning traces, using a random subset of 100

test examples. As the context window increases, the model produces more environment-applicable content and engages in more interaction turns, indicating that additional context is converted into more explicit deliberation and action attempts. However, Repair Rate and Scene Success Rate remain nearly unchanged. This decoupling between reasoning volume and repair outcome suggests a diminishing-return regime in which longer contexts primarily expand verbose or repetitive reasoning rather than improving correction quality. A likely explanation is that the bottleneck is not token budget itself, but cross-turn control quality, including state tracking, error prioritization, and action reliability under feedback. Under this view, larger windows increase opportunity for exploration but do not strengthen the policy that selects effective repairs. This also supports the context setting used in Table 2: once a sufficient reasoning budget is reached, further context expansion adds computational cost with limited performance gain.

Linking Foundational Skills to Interactive Refinement SpatialAct adopts a hierarchical task design to progressively evaluate the transition from static spatial reasoning to closed-loop scene refinement. We analyze how model capabilities evolve across the five Basic Spatial Ability tasks, Single-step Error Detection and Fix, and the primary Multi-turn Interactive Refinement task.

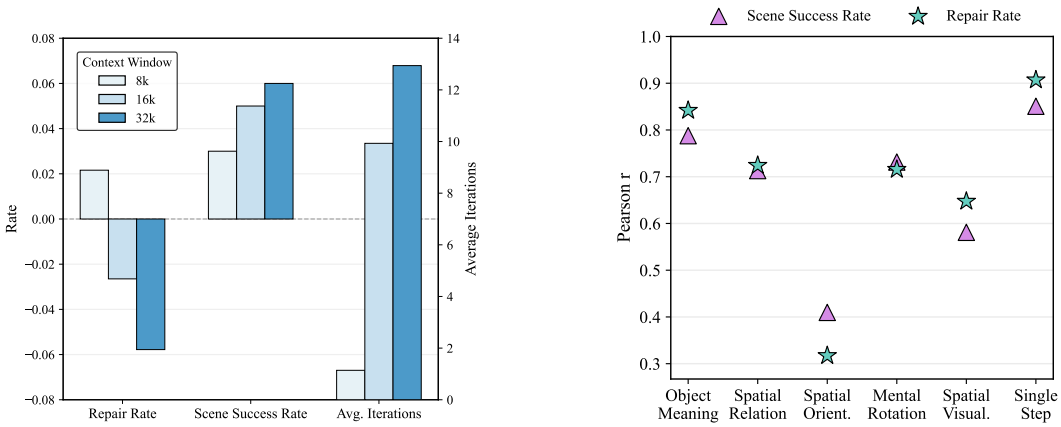
Among the five foundational tasks, Object Meaning and Spatial Relation represent the most classical forms of spatial understanding. Across both proprietary and open-source models, these tasks achieve consistently strong performance with relatively small differences between models, suggesting that current VLMs have largely acquired basic object grounding and relational reasoning abilities. Spatial Orientation introduces additional challenges by requiring alignment across different views and coordinate frames. Performance differences become more apparent in this setting. Interestingly, despite GPT-6 Astra achieving the strongest performance on most spatial tasks, it obtains only 0.623 accuracy on Spatial Orientation, lower than DeepSeek-v4.1-Pro (0.738) and Gemini-3.1 Pro (0.720). Further inspection shows that it is particularly challenged by cases requiring the camera view and the associated coordinate frame to rotate jointly.

Mental Rotation and Spatial Visualization further narrow the gap between foundational understanding and interactive refinement. Unlike purely perceptual spatial tasks, these tasks require models to predict how the scene changes after hypothetical transformations, such as object rotation, movement, or position exchange. This capability is closely related to anticipating action-induced spatial transitions, requiring models to maintain an internal representation of the scene beyond the observed configuration. Accordingly, larger performance differences emerge across models, indicating that spatial transformation reasoning is a more challenging capability than static spatial understanding.

Building on these foundational abilities, Single-step Error Detection and Fix evaluates whether models can apply spatial understanding to identify and correct a specific error in a single interaction step. This task can be viewed as a simplified version of Multi-turn Interactive Refinement, where accumulated state changes and open-ended decision making are removed. Despite this simplification, model differences become more pronounced, showing that error diagnosis and correction introduce additional challenges beyond isolated spatial understanding. We further investigate the difficulty factors of this transition in Section 3.2, where we vary error disclosure, error number, and output correction format. The results show that correction becomes substantially harder when models must identify errors without explicit hints and generate open-ended actions, with additional errors further increasing the reasoning burden.

Finally, Multi-turn Interactive Refinement represents the complete refinement setting and serves as the primary task in SpatialAct. Here, models must repeatedly diagnose spatial inconsistencies, generate corrective actions, interpret feedback, and update their understanding across interaction rounds. While frontier models perform similarly on many foundational tasks, their performance diverges substantially in multi-turn refinement. GPT-6 Astra achieves the strongest performance, with a Repair Rate of 0.681 compared with 0.073 for Qwen3.8-27B and near-zero performance for most other open-source models. This motivates our further diagnostic analysis of why models differ in multi-turn interaction and what mechanisms, including action consequence modeling and persistent spatial state maintenance, can help bridge the gap toward stronger spatial agents.

We further examine whether performance on simpler spatial tasks predicts Multi-turn Interactive Refinement across all evaluated models. As shown in Figure 13b, most foundational abilities exhibit positive correlations with both Repair Rate and Scene Success Rate, supporting the hierarchical design of SpatialAct. Among them, Single-step Error Detection and Fix shows the strongest association, with Pearson correlations of $r = 0.907$ for Repair Rate and $r = 0.851$ for Scene Success Rate, followed by Object Meaning. This suggests that reliable local error diagnosis and object grounding provide particularly important building blocks for interactive refinement. At the same time, the strength of the



(a) (b)

Figure 13: (a) Model performance under different context window settings. (b) Relationship between model performance on basic spatial tasks and complex interactive tasks, quantified by Pearson correlation coefficients (r).

relationship varies substantially across abilities. Spatial Orientation shows a much weaker association with Repair Rate, partly because it includes view-frame rotation tasks that are less directly aligned with layout correction. Overall, foundational competence is predictive of multi-turn performance, but does not fully explain it.

A.13 Controlled Single-Variable Analysis of Scene Factors

To analyze how different scene characteristics affect Multi-turn Interactive Refinement performance, we construct a controlled 45-case evaluation suite. We evaluate Gemini-3.1 Pro on this suite, where each setting varies only one designated factor while keeping other generation configurations fixed. Each controlled setting contains five scenes, enabling isolated analysis of the effects of object count, injected error count, error type, and scene domain.

Table 8 summarizes the results. Increasing the number of objects consistently reduces Repair Rate, suggesting that larger scenes introduce additional challenges for iterative correction. The number of injected errors also affects Repair Rate. In this controlled setting, scenes with higher initial error counts achieve higher Repair Rates, which may be partly explained by the larger number of opportunities for measurable error reduction. Notably, when only one error is injected, the Repair Rate is 0 despite a Scene Success Rate of 0.4. This is because Repair Rate measures the net change in the total number of errors across scenes, so errors successfully removed in fully repaired scenes can be offset by newly introduced errors in other scenes. Different error types also lead to distinct performance patterns, with Wall Conflict and Orientation Error being more challenging than Collision errors. Scene domain also affects refinement outcomes, with urban architectural scenes achieving higher Repair Rates but lower Scene Success Rates compared with indoor scenes. These results highlight that different scene characteristics influence interactive refinement performance in distinct ways.

Table 8: Controlled single-variable evaluation of scene difficulty factors. Only the designated factor varied while other generation settings are fixed in each setting.

Controlled Factor	**Setting**	**Repair Rate ↑**	**Scene Success ↑**
Number of Objects	6	0.643	0.400
	10	0.417	0.400
	12	0.308	0.200
Number of Injected Errors	1	0.000	0.400
	3	0.333	0.200
	5	0.720	0.200
Error Type	Collision	0.417	0.400
	Wall Conflict	0.231	0.200
	Orientation Error	0.111	0.200
Scene Domain	Indoor	0.417	0.400
	Urban Architectural	0.632	0.200