

---

# Appendix for OST-Bench: Evaluating the Capabilities of MLLMs in Online Spatio-temporal Scene Understanding

---

Anonymous Author(s)

Affiliation

Address

email

1	<b>A Benchmark Details</b>	<b>1</b>
2	A.1 Exploration Route Generation . . . . .	2
3	A.2 Visible Information Processing . . . . .	3
4	A.3 Rule-based Generation . . . . .	3
5	A.4 Statistics . . . . .	6
6	A.5 Benchmark Examples . . . . .	6
7	<b>B Implementation Details</b>	<b>6</b>
8	<b>C Experiment Analysis Details</b>	<b>7</b>
9	C.1 Cases of Three Error Types . . . . .	7
10	C.2 Cases of Spatio-temporal Reasoning Shortcut . . . . .	9
11	C.3 Subset Construction Process for Cross-View Analysis . . . . .	9
12	C.4 More Findings in Tables . . . . .	9
13	<b>D Inference Time of the Models</b>	<b>11</b>
14	<b>E Social Impact</b>	<b>11</b>
15	<b>F License and Access</b>	<b>11</b>
16	F.1 License and Access for Existing Assets . . . . .	11
17	F.2 License and Access for OST-Bench . . . . .	12
18	<b>A Benchmark Details</b>	
19	This section provides additional details on the construction of our benchmark, including the algorithm	
20	used for route generation, the method for determining object visibility, the rules for benchmark	
21	sample generation, and summary statistics of the generated data.	

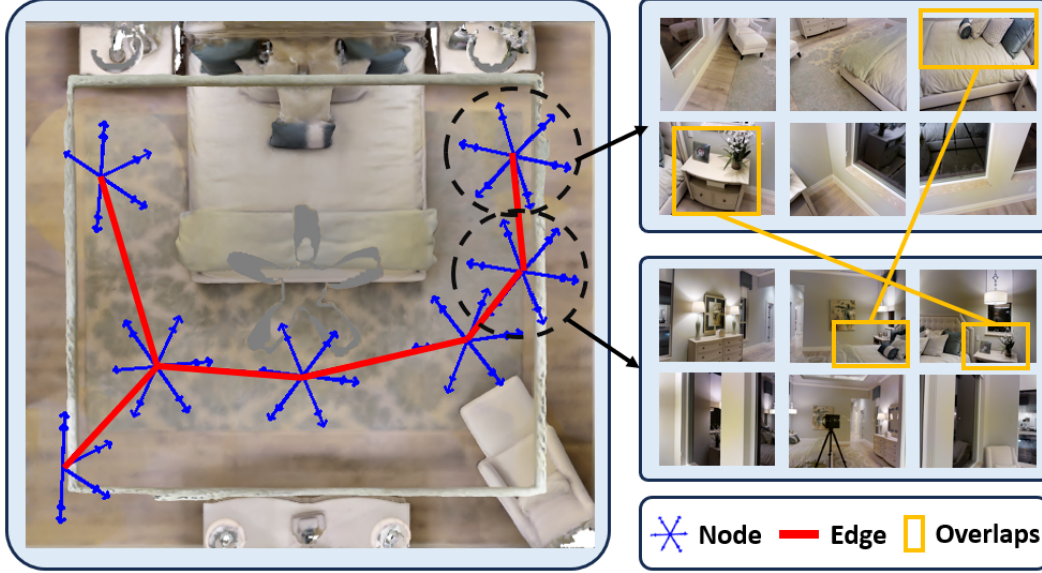


Figure 1: **Illustration of the route generation process.** The radial arrows represent multiple different viewing angles at a position, and the red edges denote connections generated by the MST algorithm. The right part shows the captured images for each viewing angle of two adjacent nodes. The agent can only move along the edges of the tree, and adjacent frames are required to have a certain amount of overlap.

## 22 A.1 Exploration Route Generation

23 While ScanNet and ARKitScenes offer egocentric video sequences with associated per-frame camera  
 24 parameters, Matterport3D provides, for each scene,  $n$  camera positions distributed throughout the  
 25 environment. From each position,  $k$  images are captured at different viewing angles, as illustrated in  
 26 Fig. 1. We aim to leverage this information to construct a simulated trajectory of an agent exploring  
 27 the scene from a first-person perspective. As mentioned in the main paper, the trajectory must  
 28 satisfy two key requirements: (a) **Path continuity**, the movement between adjacent frames should be  
 29 smooth, avoiding abrupt spatial jumps over short time intervals. (b) **Observation continuity**, adjacent  
 30 frames in the video must have a certain degree of visual overlap, which is crucial for providing the  
 31 cross-frame visual continuity necessary for constructing a coherent 3D understanding of the scene.  
 32 The videos provided by ScanNet and ARKitScenes naturally satisfy both of these requirements.

33 The video we aim to generate is a sequence of tuples  $\{(n_i, k_i, c_i)\}$ , where  $n_i$  denotes the camera  
 34 position index among the  $n$  predefined locations,  $k_i$  indicates the viewing angle index among the  
 35  $k$  available viewing angles at that position, and  $c_i$  is the corresponding captured image. Based on  
 36 the two aforementioned requirements(Fig.1), (a) We first construct a minimum spanning tree(MST)  
 37  $T(N, E)$  over all camera positions using Prim’s algorithm, where edge weights are defined by the  
 38 Euclidean distances between positions. We constrain the agent’s movement to either transitions  
 39 between neighboring positions connected by edges in the MST ( $E_{n_i, n_j} \in E$ ), or changes in viewing  
 40 angles at the same position. This design ensures path continuity throughout the simulated trajectory.  
 41 (b) We enforce that adjacent images in the sequence must have sufficient visual overlap. That is, for  
 42 any  $i \geq 0$ , the overlap between images  $c_i, c_{i+1}$  must satisfy  $Overlap(c_i, c_{i+1}) > threshold$ . This  
 43 constraint preserves observation continuity across frames. Based on these two rules, we perform a  
 44 random walk over the nodes to generate the sequence. Starting from a randomly selected initial state  
 45 with a random tuple  $(n_0, k_0, c_0)$ , at each step, we randomly select a valid and previously unseen tuple  
 46 representing the next state and append it to the sequence. This process continues until no valid tuples  
 47 remain or the sequence reaches a predefined length.

48 It is important to note that the generated videos ensure continuity in terms of paths and observations,  
 49 but do not guarantee temporal continuity (i.e., they only provide discrete frame ordering without  
 50 information on the time intervals between frames). However, since our benchmark setting uses rounds  
 51 as discrete timestamps, such temporal information is not required, and the provided data is sufficient  
 52 for our purposes.

## 53 A.2 Visible Information Processing

54 **Attribute Visibility.** For the attribute visibility of objects, to reduce computational complexity, we  
55 first apply a necessary condition: if an object is visible, then at least one of its 3D points must  
56 be projectable onto the 2D image plane within the image boundaries and without occlusion. This  
57 condition allows for the rapid elimination of most invisible objects in each image. For objects  
58 satisfying this condition, we project the surface points of their bounding boxes onto the image’s 2D  
59 plane. We first compute the projected area  $A_2$  without considering occlusion or image boundaries.  
60 Then, we calculate the visible area  $A_1$  by accounting for occlusions and restricting projections to  
61 within the image bounds. An object is deemed visible if either (1) the ratio of visible to total projected  
62 area,  $A_1/A_2$ , exceeds a predefined threshold, or (2) the absolute visible area  $A_1$  is sufficiently large.

63 **Spatial Visibility.** For the spatial visibility of objects, building on attribute visibility, we further check  
64 whether at least five vertices of the object’s 9-DoF bounding box are visible in the frames observed  
65 so far. If this condition is met, we assume the object’s center position, size (length, width, height),  
66 and related spatial information are all available, thus satisfying the criteria for spatial visibility.

## 67 A.3 Rule-based Generation

68 Our OST-Bench comprises three major categories: *Agent State*, *Agent Visible Info*, and *Agent-object*  
69 *Spatial Relationship*. Within these categories, we define a total of 15 question subtypes. Data samples  
70 are generated through a rule-based approach, guided by a set of principles outlined below.

71 (1) **Multi-round Dialogue Format.** OST-Bench adopts a multi-round dialogue setup. In each round,  
72 4–5 new frames from the video are selected sequentially in chronological order as new observations  
73 and appended to the historical observation sequence. Each question is asked at the timestamp of the  
74 last frame in the current round. All information after this timestamp is considered unavailable, and  
75 we ensure that the question is answerable based solely on the observations up to that timestamp.

76 (2) **Sample Pool Construction and Selection.** For each question subtype, we exhaustively generate  
77 all possible data samples to form a candidate pool. We ensure that no identical question-answer pair  
78 appears across different dialogue rounds (although the same question might occur, the answers must  
79 differ). In each round, we first randomly select a question subtype and then randomly select a data  
80 sample from its corresponding candidate pool as the question for that round.

81 (3) **Object Reference.** Object references in questions are divided into two types. The first is category-  
82 level reference, where a category word is used to refer to all instances of that category (e.g., “*How*  
83 *many books are there in the room?*”). The second is instance-level reference, where a specific  
84 grounded description is used to uniquely identify a single object.(e.g., “*Where is the yellow-covered*  
85 *book labeled with the word 'atomic'?*”). These descriptions are sourced from MMScan’s object-level  
86 annotations. To eliminate ambiguity, we ensure that this referred object is the only instance of its  
87 category within historical observations.

88 (4) **Memory-based Reasoning Requirement.** To rigorously test a model’s ability to reason over  
89 long-term memory and avoid overly simple questions, we ensure that no question can be answered  
90 using only the newly added observations in the current round. Each question requires integrating  
91 information from both the current and previous dialogue rounds. For example, we ensure that at least  
92 one relevant object is absent from the observations in the current round, thereby requiring the model  
93 to recall it from prior rounds.

94 (5) **Ensuring Clarity and Avoiding Ambiguity.** To ensure the validity and clarity of the questions and  
95 to avoid controversial or ambiguous cases, we impose specific thresholds during sample generation  
96 so that the answers are unambiguous and clearly inferable. For example, when a question involves  
97 comparing two distances, we require the difference between the distances to exceed a predefined  
98 threshold to ensure a significant contrast. Similarly, for questions such as determining whether an  
99 object is on the left or right, we require the object to be clearly positioned on one side. Objects  
100 located near the decision boundary (e.g., close to the center) are excluded to prevent ambiguity in  
101 interpretation.

102 Fig.2 presents the predefined templates used for generating questions across different subtypes. The  
103 specific generation strategies for each subtype are detailed below:

Agent State	Position(JUD.)	Q: Assuming the direction at the end of {round ID} is forward, did you move a certain distance <i>left</i> or <i>right</i> / <i>forward</i> or <i>backward</i> from that position? O: [left, right] / [forward, backward]
	Position(EST.)	Q: How <i>far</i> is your current position from where you were at the end of {round ID}? (in meters)
	Orientation(JUD.)	Q: Using your orientation at the end of {round ID} as a reference, has your current orientation rotated <i>clockwise</i> or <i>counterclockwise</i> by a certain angle (<180) relative to that orientation? O: [clockwise, counterclockwise]
	Orientation(EST.)	Q: Using your orientation at the end of {round ID} as a reference, how many <i>degrees</i> has your current orientation rotated <i>clockwise/counterclockwise</i> relative to the previous orientation?
Agent-object Spatial	Distance(JUD.)	Q1: {object1}, {object2}, and {object3}, which one is the <i>closest to/farthest from</i> you now? O1: [{object1}, {object2}, {object3}] Q2: Compared to the end of {round ID}, are you now <i>closer or farther away</i> from {object}? O2: [closer, farther] Q3: Compared to the end of {round ID}, are you now <i>closer to or farther away</i> from {object1}/{object2}?
	Distance(TEMP.)	Q: In which round were you <i>closest to/farthest from</i> {object}?
	Distance(EST.)	Q: Please recall {object}, what is the horizontal <i>distance</i> between you and this object now (in meters)?
	Distance(JUD.)	Q1: Is the {object} to your <i>left/right</i> now? O1: [left, right] Q2: Which direction is {object} to you now: <i>front left, front right, rear left, or rear right</i> ? O2: [front left, front right, rear left, rear right] Q3: Which two objects are on the <i>same side</i> of you now? {object1}, {object2}, and {object3}.
	Distance(TEMP.)	Q: At the end of which round were both of {object1} and {object2} on your <i>left side</i> ?
	Distance(EST.)	Q: Based on your current orientation, in what ( <i>counter</i> ) <i>clockwise direction</i> (in degrees) is {object} from your position?
Agent Visible Info	Existence(JUD.)	Q: Remember, have you <i>seen</i> any {object type} so far? O: [Yes, No]
	Existence(TEMP.)	Q1: When did you <i>first discover/last see</i> {object} (index of the turn)? Q2: In which round did you <i>see both</i> {object1} and {object2} simultaneously?
	Quantity(CNT.)	Q: Remember, <i>how many</i> {object type}(s) have you <i>seen</i> so far?
	Diversity(JUD.)	Q: Which one was <i>newly discovered</i> in this round, {object1}, {object2} or {object3} ? O: [{object1}, {object2}, {object3}]
	Order(JUD.)	Q: What will be the <i>first-time appearance order</i> of {object type1}, {object type2} and {object type3}? O: [{order1}, {order2}, {order3}, {order4}]

Figure 2: **Rule-based generation templates for all subtypes in OST-Bench.** Placeholders to be filled with specific content are marked in red, and question focal points are highlighted in blue. "JUD." / "CNT." / "TEMP." / "EST." are abbreviations for "judgement", "counting", "temporal-localization", and "estimation"; "Q" and "O" denote "Question" and "Options"

**Agent State.** This category encompasses tasks that require the agent to judge or estimate its own spatial state, including its position and orientation. Since there is no globally defined coordinate system in OST-Bench, all measurements are made relative to a specific historical time point.

- *Position (Judgement)*: In this type, the task is to determine whether the agent has moved to the left or right (forward or backward), relative to its position and orientation at the end of a previous round  $T_1$ . The question is formulated as a binary choice, with the correct answer being either *left* or *right*(*forward* or *backward*). Let  $P_1$  and  $O_1$  denote the position and orientation at the end of round  $T_1$ , and  $P_2$  denote the current position. We compute the parallel and perpendicular components of the vector  $P_2 - P_1$  with respect to  $O_1$ . A question is generated only if the absolute value of either component exceeds a predefined threshold (1 meter). The correct answer is determined by the sign of the respective component: a positive value indicates forward or right, while a negative value indicates backward or left.
- *Position (Estimation)*: In this subtype, the task is to estimate how far the agent has moved from its position at the end of a previous round  $T_1$ . The ground-truth answer is defined as the Euclidean distance between the agent's current position and its position at  $T_1$ .
- *Orientation (Judgement)*: This binary-choice question asks whether the agent has rotated clockwise or counterclockwise by an angle(less than 180 degrees) relative to its orientation at the end of round  $T_1$ . We compute the angle between the current orientation vector and the one at the end of  $T_1$ . To exclude ambiguous borderline cases, questions are generated only

123 if the angle lies within the intervals  $[\theta, 180 - \theta]$  or  $[180 + \theta, 360 - \theta]$ , where  $\theta$  is a threshold  
 124 used to exclude borderline cases. Angles within the first interval indicate clockwise rotation,  
 125 while those within the second indicate counterclockwise rotation.

- 126 • *Orientation (Estimation)*: In this question type, the task is to estimate how many degrees the  
 127 agent has rotated, clockwise or counterclockwise, relative to its orientation at the end of a  
 128 previous round  $T_1$ . The answer is given as the angle between the current orientation and the  
 129 orientation at the end of round  $T_1$ .

130 **Agent Visible Info.** All objects involved in this category of questions must satisfy the attribute  
 131 visibility constraint, meaning that their existence must be identifiable from past observations. This  
 132 category evaluates the model’s understanding of agent visible information, including subtasks such  
 133 as object existence, quantity, diversity, and the order of appearances.

- 134 • *Existence (Judgement)*: This type asks whether a certain category was visible in any of the  
 135 previous observations. The answer is binary: *yes* or *no*. To balance positive and negative  
 136 samples, we generate questions for object categories that do not appear in prior observations  
 137 with a 50% probability.
- 138 • *Existence (Temporal Localization)*: This type includes two forms of queries: (1) Identifying  
 139 the earliest/latest round in which a specific object was visible; (2) Identifying the round  
 140 in which two specific objects were simultaneously visible. For both forms of queries, we  
 141 ensure the answer is unique—i.e., there is exactly one round that satisfies the condition.
- 142 • *Quantity (Counting)*: This task requires counting how many objects of a specified category  
 143 were visible in past observations. To avoid trivial cases, we exclude questions where the  
 144 correct answer is one. Additionally, to balance the distribution, negative samples—where  
 145 the target category does not appear at all—are introduced to constitute 25% of the total  
 146 samples.
- 147 • *Diversity (Judgement)*: This question type asks which object is newly observed in the current  
 148 round. The agent must choose one object from three candidates, all of which are visible in  
 149 the current observation. Among them, only one has not appeared in any previous round,  
 150 while the other two have been seen before.
- 151 • *Order (Judgement)*: This question type involves determining the appearance order of three  
 152 different object categories. The agent must select the correct sequence from four given  
 153 permutations. We ensure that the first appearance round of each object category is distinct  
 154 to avoid ambiguity in ordering.

155 **Agent-Object Spatial Relationship.** This category focuses on constructing spatial metric relation-  
 156 ships between the agent and a specific object  $O$  at a specific time  $T$ . The distance between the agent  
 157 and object  $O$  at time  $T$  is defined as the shortest distance from the camera coordinate to any point  
 158 in the object’s point cloud. The angle of object  $O$  relative to the agent at time  $T$  is computed as the  
 159 angle between the camera’s horizontal orientation vector and the vector pointing from the camera  
 160 to the center of object  $O$ . All objects involved in this category must satisfy the spatial visibility  
 161 constraint, which means that their center coordinates, dimensions (length, width, height), and other  
 162 spatial properties must be reliably obtainable from previous observations.

- 163 • *Distance (Judgement)*: This question type includes three forms of queries: (1) determining  
 164 which of the three objects is currently farthest from or closest to the agent; (2) judging  
 165 whether the current distance between the agent and a specific object is greater or smaller  
 166 than the distance at the end of a previous round; (3) judging whether the current distances  
 167 between the agent and two specific objects are greater or smaller than those at the end of a  
 168 previous round, with four possible answer choices. For the first form, at least one object  
 169 must be invisible in the current round, and the distance to the correct answer object must  
 170 differ significantly (i.e., by more than a predefined threshold) from the distances to the other  
 171 two objects. For the second and third forms, the change in distance between the two time  
 172 points must also exceed the threshold to ensure a meaningful distinction.
- 173 • *Distance (Temporal Localization)*: This task asks the agent to identify the round in which it  
 174 was closest to or farthest from a specific object. The distance in the correct round must be  
 175 significantly smaller (for closest) or larger (for farthest) than in all other rounds.

- *Distance (Estimation)*: This query requires estimating the current distance between the agent and a specific object, which is invisible in the current round and thus requires recalling information from previous rounds.
- *Direction (Judgement)*: This question type includes three forms of queries: (1) judging whether a specific object is currently on the agent’s left or right side; (2) judging whether a specific object currently lies in the left-front, left-back, right-front, or right-back quadrant relative to the agent; (3) identifying which two out of three objects are currently on the same side of the agent. For the first two forms, we enforce angular thresholds by excluding objects whose relative angles fall within 10 degrees of the decision boundaries between sides or quadrants, thereby avoiding ambiguity. For the third form, at least two of the three objects are invisible in the current round, forcing the model to rely on memory.
- *Direction (Temporal Localization)*: This query asks the agent to identify the round in which both objects A and B were located on the same side (left or right) relative to the agent. We ensure that in each round, both objects are clearly on either the left or right side (at least 10 degrees away from the decision boundary), and that there is exactly one round satisfying this condition.
- *Direction (Estimation)*: This query requires estimating the angle, clockwise or counterclockwise, of a specific object relative to the agent’s current orientation. The object is not visible in the current round, requiring retrieval from prior observations.

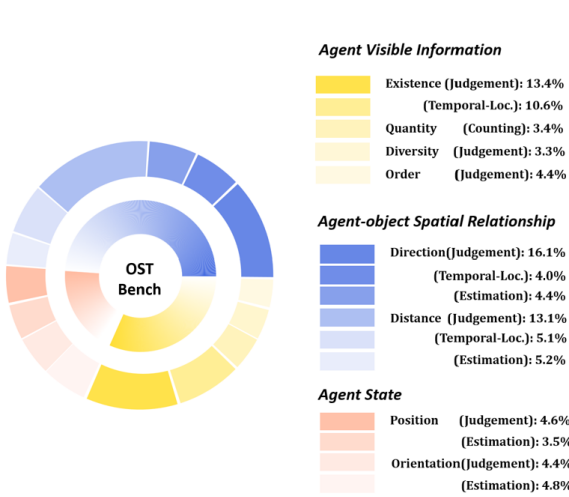


Figure 3: Distribution of sample counts across different subtypes in OST-Bench.

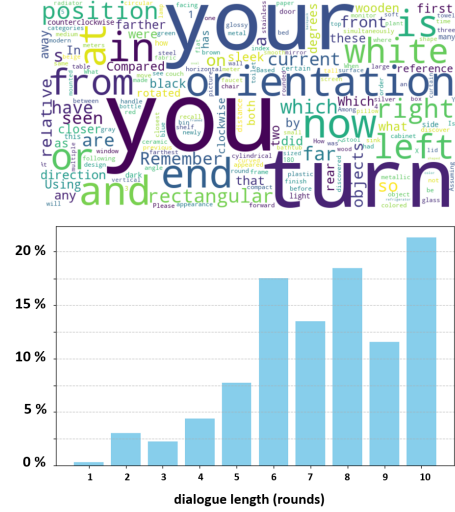


Figure 4: Word cloud (top) and dialogue length distribution (bottom) of OST-Bench.

#### 195 A.4 Statistics

196 Based on the generation methods described above, OST-Bench totally consists of 1.4k trajectories(a  
 197 trajectory per scene) and 10k data samples. The distribution of sample counts across different  
 198 subtypes is shown in Fig. 3. We also present in Fig. 4 the word frequency distribution in OST-Bench  
 199 (visualized as a word cloud), as well as the distribution of dialogue lengths.

#### 200 A.5 Benchmark Examples

201 In Fig. 9 and 10 we provide more examples from our benchmark, including a total of 12 data samples  
 202 from two scenes (exploration trajectories).

### 203 B Implementation Details

204 For the multi-round dialogue, we first provide a system prompt to inform the models of the task setup.  
 205 In each round, we sequentially input a set of images representing new video frames, along with a

<b>System Prompt</b>	<p>"Assume you are currently exploring a room where all objects are stationary. Over time, you change your position and orientation within the room and take images.</p> <p>Now, I will engage you in a multi-round dialogue (a total of {num of rounds} ). In each round, I will provide you with {num of images per round} images taken from the beginning to the end of that round. Please answer my questions based on your state(position/orientation) at each round's end (last image)."</p>	
<b>User message</b>	<p>&lt;image&gt; + "For the {round ID}, these are the {num of images per round} images in chronological order. The question for this turn is: {question}. To answer this question, you need to combine information from past rounds. Please give me your answer and reason in a JSON format."</p>	
	<i>Judgement</i>	Please choose the answer from {options} .
	<i>Counting/Temporal-Loc/Estimation</i>	Please provide a numerical value as the result. The information I provided is sufficient for you to infer the value; do not refuse to answer!

Figure 5: Model input content, including the system prompt and inputs for each round. Text placeholders to be filled are highlighted in red, while the green <image> token represent image placeholders to be filled.

prompt containing a question, as illustrated in Fig.5. For judgment questions, we include the options in the prompt. For the other three question formats (estimation, counting, and temporal-localization), we prompt the model to output a specific numerical value and explicitly instruct it to answer the question. This instruction is necessary, as we observed during experiments that models may otherwise refuse to respond, claiming insufficient information.

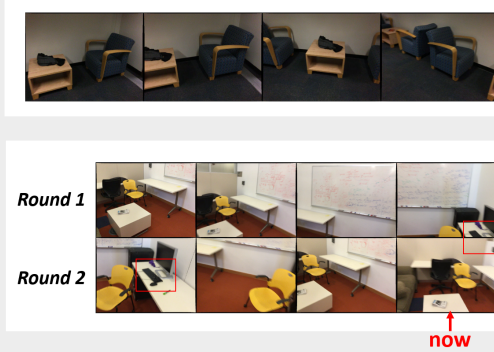
For proprietary models, we interact with the OpenAI and Anthropic APIs, both of which support multi-round dialogue with image inputs. In these APIs, models are invoked by explicitly specifying their model names. For the OpenAI API, we use *gpt-4o* for GPT-4o, *gpt-4.1* for GPT-4.1, *gemini-2.0-flash* for Gemini-2.0-Flash, and *gemini-2.0-flash-thinking-exp* for its thinking variant. For the Anthropic API, we use *claude-3-5-sonnet-latest* to access Claude-3.5-Sonnet. The system prompt is set to the task description, and each round’s input includes newly added images and questions. For open-source models (InternVL, QwenVL, LLaVA-Onevision, and LLaVA-Video), we manually construct the multi-round context by concatenating the dialogue history, new images, and the current prompt as the input at each round. To avoid out-of-memory errors, input images are resized accordingly. For models with up to 8 billion parameters, inference is run on a single NVIDIA A100 GPU. For models with 32 billion parameters or more, we perform multi-GPU inference using 8 NVIDIA A100 GPUs via model and data parallelism. Additionally, we implement multithreaded processing to accelerate the inference of open-source models.

## C Experiment Analysis Details

### C.1 Cases of Three Error Types

In Fig.6, we present examples of the three types of errors: *Prompt Analysis Error*, *Perception Error*, and *Reasoning Error*. In the first example of Prompt Analysis Error, the prompt explicitly requires the model to output a specific quantity. However, the model fails to interpret this requirement correctly and responds with "no" instead of providing a numerical answer such as "0". In the second example, the model misunderstands the meaning of the word "discover" in the prompt. It assumes that partially seeing the keyboard in Round 1 does not count as a discovery and that only fully observing it in Round 2 qualifies as such. This misinterpretation leads to an incorrect answer. In the two Perception Error examples, the model fails to correctly identify a washbin located in the corner of the room and only detects one of the two lamps in the bedroom, missing the other. In the Reasoning Error examples, although the model correctly understands the prompt and accurately perceives the location of the target objects, it makes an error in reasoning about their spatial relation with the agent, leading to incorrect conclusions.

## Prompt Analysis Error



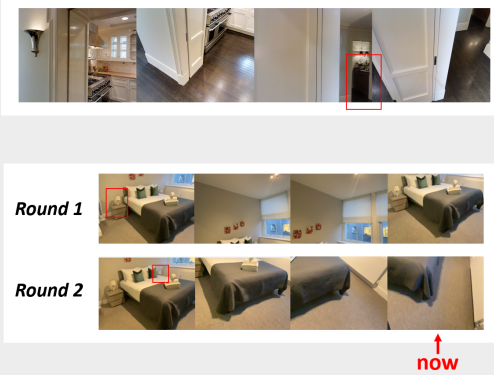
How many **backpacks** have you seen so far?

No, there are no backpacks. ❌

At which round did you first discover the **keyboard**?

Round 2. In the last picture of the first round, I could roughly see an object resembling a keyboard; in the first picture of the second round, I could clearly identify it as a keyboard. *Since I couldn't determine whether it was a keyboard when I saw it in the first round, I only discovered the keyboard for the first time in the second round.* ❌

## Perception Error



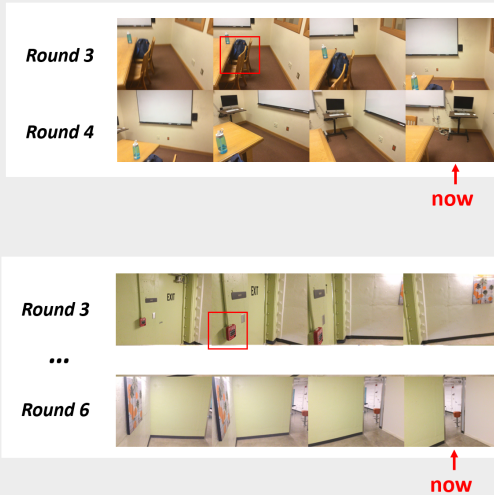
Remember, have you seen any **washbasins** so far?

No. Combining information from all four images, there is **no washbasin** visible in any of them. ❌

How many **lamps** have you seen so far?

One. I **only see one** lamp on the stand in image-1 in round 1 and there are no other lamps in the other images. ❌

## Reasoning Error



In which round were you farthest from the **dark blue backpack**?

Round 4. In Round 3, the backpack was clearly visible on the chair near the table in all images. In Round 4, the focus shifts towards the whiteboard and the computer, *moving the backpack out of view, indicating a greater distance.* ❌

Is the **red wall-mounted fire extinguisher** to your left/right now?

Left. In Round 3, I moved from facing the EXIT door and fire extinguisher to facing the wall with the picture, placing the extinguisher to my left. In Round 4-5, I continued forward along the wall with the picture. In Round 6, I've moved further along that path, turning to reveal a new room. *Given the pathway followed and my earlier position, the fire extinguisher remains on my left.* ❌

Figure 6: Illustrative Examples of the Three Error Types: Prompt Analysis Error, Perception Error, and Reasoning Error.

## C.2 Cases of Spatio-temporal Reasoning Shortcut

In the main paper, we have discussed the *Spatio-temporal Reasoning Shortcut* phenomenon exhibited by the models. In Fig.7, we provide additional examples to further demonstrate the prevalence of this behavior. For clarity, we display only the key video frames relevant to each question. Temporal expressions in the questions and model responses are replaced with t1, t2, and t3, and marked above the corresponding frames. All of these examples demonstrate the model’s tendency to rely on shortcuts in spatio-temporal reasoning.

In the first example, GPT-4o incorrectly infers that the blackboard has moved closer simply based on its transition from being invisible to visible, ignoring spatial cues such as the chairs and the decorations on the wall. In the second example, Gemini-2.0-Flash performs a seemingly correct inference using only two frames (the current and target frames), concluding that the wall currently in front of the agent is adjacent and perpendicular to the wall in t1, while disregarding intermediate frames that contain crucial contradictory evidence. In the third example, InternVL-2.5-78B observes that the TV was on the right side of the room in earlier frames and then directly assumes it remains there when it becomes invisible. In the fourth and fifth examples, the models make incorrect judgments due to the target object being invisible in the specific frames. In the sixth example, the model only focuses on the frames where the stand appears and the current frame, while skipping over intermediate frames that indicate the agent turned around, wrongly assuming that the current orientation is aligned with the previous one.

## C.3 Subset Construction Process for Cross-View Analysis

As mentioned in the main paper, when constructing the dataset for the Cross-View subset, we first generate an initial batch of data using a rule-based method and then manually filter the data to obtain the final set of 200 samples. Our rule-based construction method for generating the Cross-view subset with different levels of difficulty is described as follows:

(a) **Single-Step Spatial Connection.** We first iterate over all possible object pairs  $(O_1, O_2)$  in the scene. For each object pair, we traverse all possible frame pairs  $(F_1, F_2)$  within the video sequence. A frame pair is selected if it satisfies the following conditions: (1)  $O_1$  is visible in  $F_1$  but not in  $F_2$ ; (2)  $O_2$  is visible in  $F_2$  but not in  $F_1$ ; (3)  $F_1$  and  $F_2$  share at least one overlapping object. This setup ensures that the spatial relationship between  $O_1$  and  $O_2$  can be inferred via single-step reasoning. All tuples  $(O_1, O_2, F_1, F_2)$  satisfying these constraints are collected as initial data for the **keyframe-based context**. To construct the **sequence-based context**, we embed  $F_1$  and  $F_2$  into a video sequence  $V$  that includes frames not containing  $O_1$  or  $O_2$ , resulting in tuples of the form  $(O_1, O_2, V)$ .

(b) **Multi-Step Spatial Connection.** Similarly, we iterate over all object pairs  $(O_1, O_2)$  and traverse all frame triplets  $(F_1, F_2, F_3)$  from the video sequence. A triplet is selected if it meets the following conditions: (1)  $O_1$  is visible in  $F_1$  but not in  $F_2$  or  $F_3$ ; (2)  $O_2$  is visible in  $F_2$  but not in  $F_1$  or  $F_3$ ; (3)  $F_1$  or  $F_3$  share at least one overlapping object; (4)  $F_2$  or  $F_3$  share at least one overlapping object; (5)  $F_1$  and  $F_2$  have no overlapping objects. This configuration ensures that solving the problem requires multi-step reasoning. All valid tuples  $(O_1, O_2, F_1, F_2, F_3)$  satisfying these constraints are collected as initial data for the **keyframe-based context**. Similarly, to construct the **sequence-based context**, we embed  $F_1, F_2$  and  $F_3$  into a video sequence  $V$  that includes frames not containing  $O_1$  or  $O_2$ , resulting in tuples of the form  $(O_1, O_2, V)$ .

## C.4 More Findings in Tables

**Difficulty of Estimation Tasks.** As shown in Table 2 in the main paper, models perform particularly poorly on estimation tasks, achieving scores well below the chance-level baseline. Humans also struggle with these questions, obtaining significantly lower scores compared to other task categories. This is because estimation questions go beyond innate human perceptual abilities. Humans are better at perceiving spatial relationships approximately than estimating spatial measurements precisely, requiring not only spatial reasoning but also extensive empirical knowledge accumulated from experience.

**Detection Success vs. Counting Failure.** As shown in Table 2 in the main paper, models achieve notably high scores on object-existence questions, demonstrating a strong ability to identify whether and when objects appear. However, their performance drops significantly for object-quantity tasks,

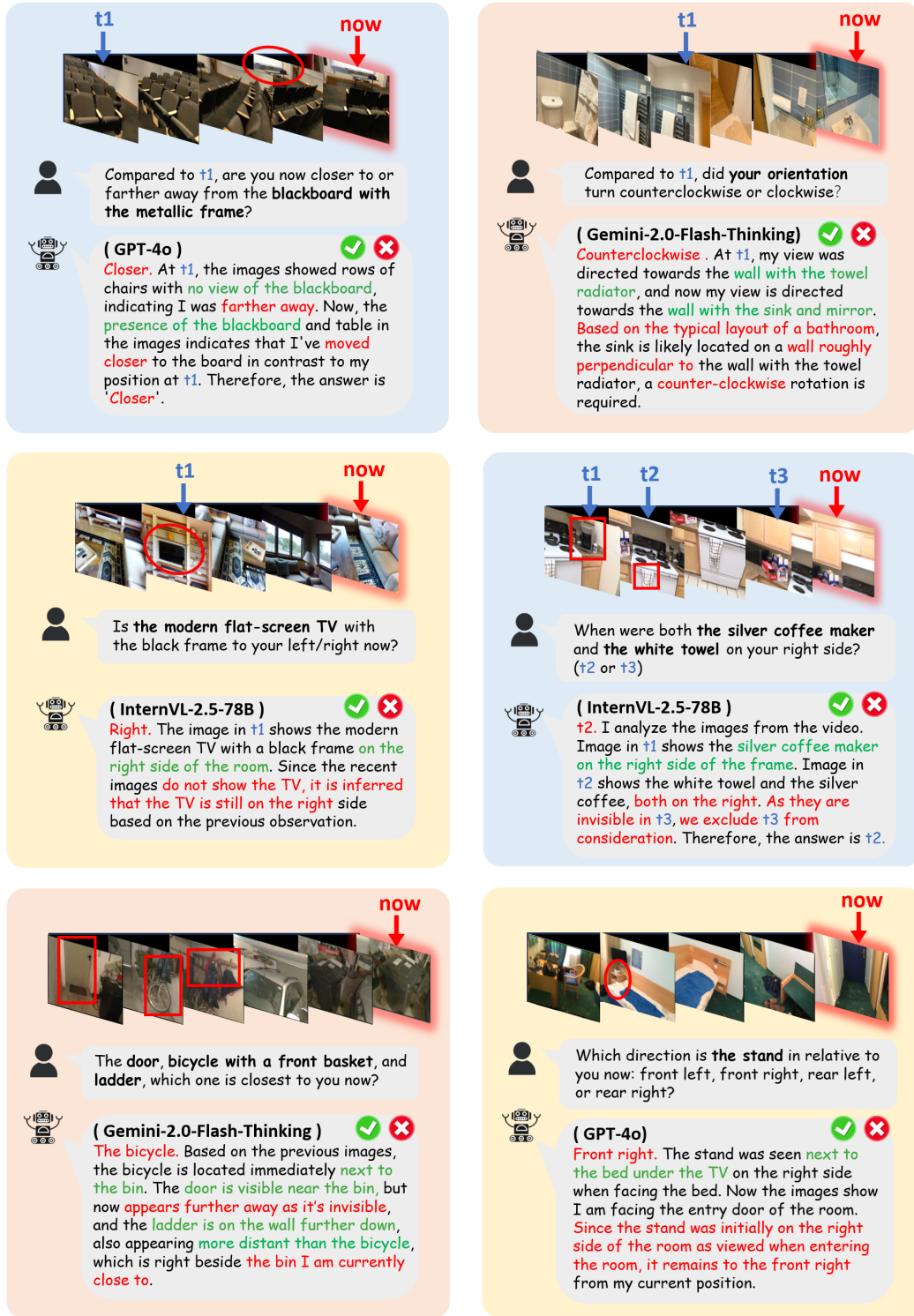


Figure 7: More examples of Spatio-temporal Reasoning Shortcuts. Green text marks correct reasoning; red indicates errors. For clarity, only key video frames relevant to each question, with temporal references replaced by  $t_1$ ,  $t_2$ , and  $t_3$ .

290 which require counting. Upon examining specific cases, we found that models frequently confuse  
291 whether objects across frames are the same or distinct, mistaking two different objects as identical or  
292 failing to track the same object across frames. This suggests that the task demands not just detection  
293 capabilities but also cross-frame reasoning.

294 **The Illusion of Better Distance Understanding.** As shown in Table 2 in the main paper, models  
295 appear to perform slightly better on Agent-object distance questions compared to Agent-object  
296 direction, but this advantage is superficial. This is primarily due to the *Spatio-temporal Reasoning*  
297 *Shortcut* phenomenon: models tend to assume that objects currently visible are closer, while those  
298 out of view are farther away, without engaging in genuine spatial reasoning. Although this heuristic  
299 can occasionally lead to correct answers, since such patterns do occur in a small portion of our  
300 benchmark, it fails to generalize. As a result, models still perform poorly on Agent-object distance  
301 questions overall.

## 302 D Inference Time of the Models

303 Although OST-Bench does not impose real-time constraints, we conducted a supplementary study on  
304 models’ inference time, indirectly reflecting the delay in decision-making exhibited by the models in  
305 real-world embodied tasks. Since the inference time of proprietary models is also affected by network  
306 latency, we restrict our analysis to open-source models and report their inference time per question.

307 The Fig.8 illustrates how the model’s inference time per question changes as the duration of explo-  
308 ration increases. The results reveal a clear trend: as exploration time increases and more historical  
309 context accumulates, inference latency grows rapidly. When the number of dialogue rounds becomes  
310 large (e.g., beyond 10), the inference time becomes prohibitively high, especially for large-scale  
311 models, making real-time interaction impractical. This latency surge stems from the fact that any  
312 frame in history may contain critical information, forcing the model to attend to a growing number of  
313 frames at every step. Thus, inference time scales approximately linearly with history length.

314 To provide context, we also measured human inference time. While average latency isn’t directly  
315 comparable due to individual variation, we find that for all human testers, response time remained  
316 stable regardless of how long the exploration had lasted. This starkly contrasts with model behavior.  
317 The underlying reason is that humans can actively abstract and compress information throughout  
318 the exploration process, forming an internal knowledge base. Rather than treating each question as  
319 a fresh input, humans recall previously formed abstractions, enabling efficient reasoning without  
320 reprocessing all historical data.

321 This comparison highlights a critical need: for models to perform well in real-world embodied tasks,  
322 they must learn to dynamically distill and retain knowledge during exploration. Instead of passively  
323 accumulating history or answering questions in isolation, models should develop mechanisms to  
324 summarize and store essential information in an efficient, retrievable form, paving the way for scalable  
325 and real-time embodied reasoning.

## 326 E Social Impact

327 OST-Bench aims to advance the development of multimodal large language models (MLLMs) with  
328 stronger online spatio-temporal reasoning capabilities, which are critical for real-world embodied  
329 tasks such as assistive robotics, autonomous navigation, and human-robot interaction. By introducing  
330 a more realistic and challenging benchmark, we hope to drive progress toward more reliable and  
331 generalizable agents capable of perceiving and reasoning in real-world environments under online  
332 settings. However, as the benchmark assumes a static environment and focuses only on perception  
333 and reasoning, there is a risk of overestimating model readiness for real deployment. Caution is  
334 needed to avoid misuse or overreliance on models without broader capabilities like interaction or  
335 manipulation, which are essential for safe and responsible AI integration in the real world.

## 336 F License and Access

### 337 F.1 License and Access for Existing Assets

338 As mentioned in the main paper, our real-world scene data is sourced from ScanNet, Matterport3D,  
339 and ARKitScenes. To access and use these three datasets, users should follow their original licenses

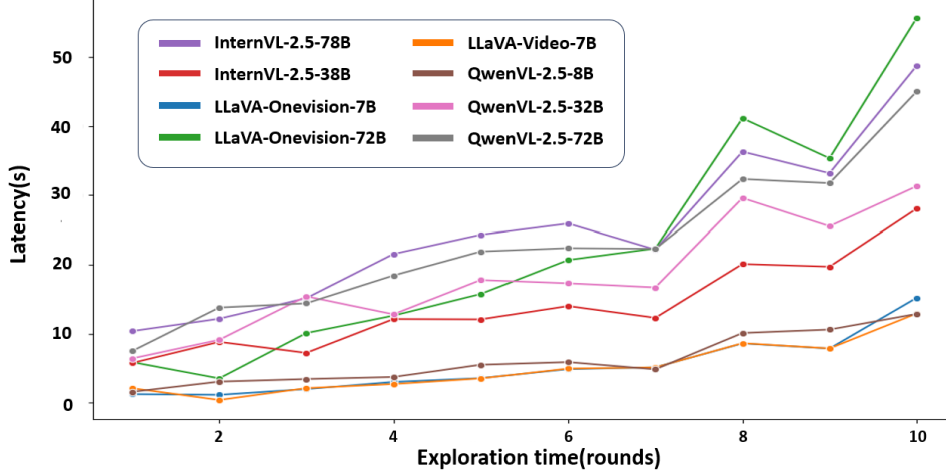


Figure 8: The trend of the model’s inference time per question as the duration of exploration increases.

[4, 3, 1], and ask their official hosts for authorization. Additionally, our annotated data come from EmbodiedScan and MMScan, access to these datasets requires submitting a request via a Google Form [2] and following the license attached to the form.

We use ScanNet, Matterport3D, and ARKitScenes as the scene data and leverage the video information provided in them. We adopt the bounding box annotations and textual annotations from EmbodiedScan and MMScan as the base datasets for our benchmark. Throughout the usage of these datasets, their licenses and terms of use are properly respected.

## F.2 License and Access for OST-Bench

The OST-Bench dataset is distributed under the Creative Commons Attribution 4.0 International License (CC BY 4.0) and available for direct download at <https://github.com/rbler1234/OST-Bench> or <https://www.kaggle.com/datasets/jinglilin/ost-bench/data>.

We release our benchmark under the CC-BY license and Terms of Use, and require that any use of the dataset for model evaluation be properly disclosed. This license supplements but does not override the original licenses of source materials; users must also comply with all relevant legal requirements concerning data subjects. This statement clarifies the obligations and liabilities associated with using this benchmark. While we strive to ensure the accuracy and legality of all samples, we do not guarantee their absolute completeness or correctness. We assume no responsibility for any legal or other issues that may arise from the use of OST-Bench, including but not limited to copyright infringement, privacy violations, or the misuse of sensitive information. By accessing, downloading, or using OST-Bench, you acknowledge that you accept this statement and agree to comply with the full terms of the CC-BY license. If you do not agree with these terms or the CC-BY license, you are not permitted to use this benchmark. OST-Bench will be hosted and maintained on GitHub and the Kaggle platforms.

## References

- [1] Arkitscenes license. <https://github.com/apple/ARKitScenes/blob/main/LICENSE>.
- [2] Embodiedscan and mmscan access. [https://docs.google.com/forms/d/e/1FAIpQLScUXEDTksGiqHZp31j7Zp7z1CNV7p\\_08uVwP\\_Nbzfn3g6hhw/viewform](https://docs.google.com/forms/d/e/1FAIpQLScUXEDTksGiqHZp31j7Zp7z1CNV7p_08uVwP_Nbzfn3g6hhw/viewform).
- [3] Matterport3d license. [https://kaldir.vc.in.tum.de/matterport/MP\\_TOS.pdf](https://kaldir.vc.in.tum.de/matterport/MP_TOS.pdf).
- [4] Scannet license. [https://kaldir.vc.in.tum.de/scannet/ScanNet\\_TOS.pdf](https://kaldir.vc.in.tum.de/scannet/ScanNet_TOS.pdf).



Figure 9: **Example 1 of OST-Bench data samples.** Each row represents the newly added observations in each round, with images input from left to right within each round. The example shows the question-answer pairs from the first six rounds.

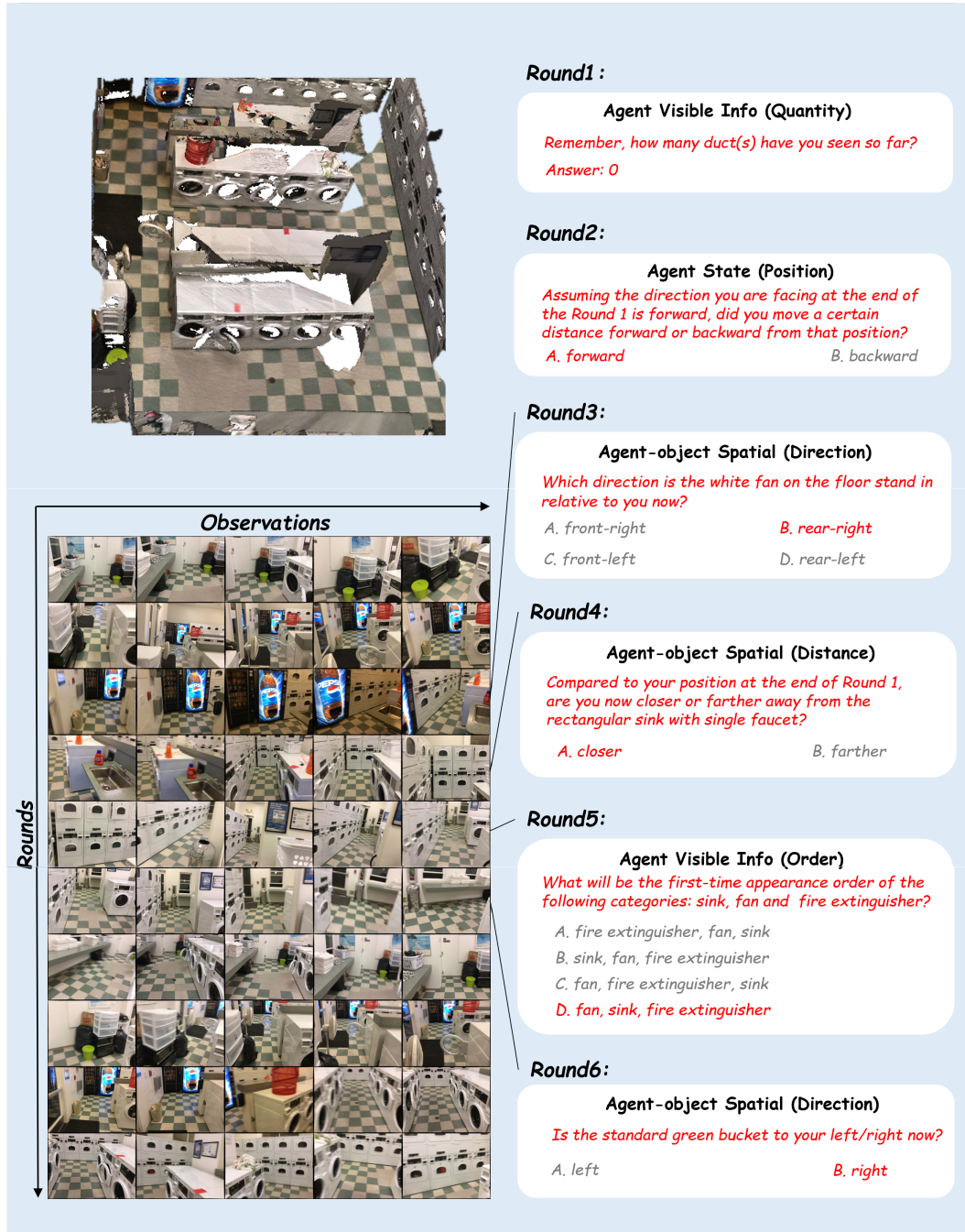


Figure 10: **Example 2 of OST-Bench data samples.** Each row represents the newly added observations in each round, with images input from left to right within each round. The example shows the question-answer pairs from the first six rounds.