

---

# Situat3DChange: Situated 3D Change Understanding Dataset for Multimodal Large Language Model (Supplementary Materials)

---

Ruiping Liu<sup>1</sup> Junwei Zheng<sup>1</sup> Yufan Chen<sup>1</sup> Zirui Wang<sup>1</sup> Kunyu Peng<sup>1</sup>  
Kailun Yang<sup>2</sup> Jiaming Zhang<sup>1,3,\*</sup> Marc Pollefeys<sup>3</sup> Rainer Stiefelhofen<sup>1</sup>  
<sup>1</sup> Karlsruhe Institute of Technology (KIT) <sup>2</sup> Hunan University <sup>3</sup> ETH Zurich

## A Data Generation

The data generation process includes situation sampling, long-form text generation, query generation for the long-form text, and QA generation. It is based on human observations of changes, object attributes, and allocentric object relationships in 3DSSG [9], as well as egocentric relationships between the human and the objects.

### A.1 Situation Sampling

We follow the situation categories of MSQA [4], namely sitting, interacting, and standing, but with more detailed geometric analysis:

**Sitting.** The 28 seat categories in 3RScan [8] are grouped into four types: 3 large seats with backrests (*e.g.*, sofa), 16 small seats with backrests (*e.g.*, armchair), 1 large seat without a backrest (bed), and 8 small seats without backrests (*e.g.*, beanbag). Seatable and backrest areas are classified by surface normals, or by nearby walls within 0.5 m if no backrest exists. For small seats, the seating point is the bounding box center, oriented away from the backrest. For large seats, we select a point with a backrest behind and open space (0.5–1 m) in front. If no backrest or wall is present, the seat center is used, facing the room center.

**Sitting on couch\_2**



**Location:** (4.63, -1.27, 0.12)  
**Orientation:** (0.77, -0.64, 0.07)  
**Descriptive Situation:**  
Sitting on the couch with a kitchen cabinet to the right.

**Standing with chair\_11 3 o'clock**



**Location:** (0.91, 0.71, 0.06)  
**Orientation:** (0.92, -0.39, -0.03)  
**Descriptive Situation:**  
Opening the kitchen cabinet in front, with a wooden chair to the right within arm reach.



**Interacting with curtain\_21**



**Location:** (4.63, -1.27, 0.11)  
**Orientation:** (0.77, -0.64, 0.07)  
**Descriptive Situation:**  
Adjusting the curtain in front, with a couch behind me.

**Interacting with vacuum cleaner\_29**



**Location:** (1.05, -1.49, 0.16)  
**Orientation:** (-0.31, -0.81, -0.50)  
**Descriptive Situation:**  
Cleaning with the vacuum cleaner in front, a wooden TV stand to the left within arm reach.

Figure 1: Examples of **sitting**, **standing**, and **interacting** situations. Each includes the location, orientation, egocentric view, panorama, and descriptive situations based on the scene’s holistic context.

---

\*Corresponding author.

**Interacting.** We consider objects with a dominant horizontal normal as interactable. A point is randomly selected from the standable floor regions, oriented toward the object center and within 5 degrees of the object’s dominant normal, at a distance of 0.3 to 0.5 m from its bounding box. **Standing.** The standing situation is anchored to the nearest object. In scenes with few objects (*e.g.*, a stairwell), multiple situations may share the same anchor object but differ in their proximity orientations, such as 3 o’clock (right) and 5 o’clock (back).

The aforementioned process generates a brief situation with an anchor object, location, and orientation, as shown in Fig. 1. The egocentric view and panorama are captured to represent the perspective of a wearable device or embodied agent. Since situations anchored solely to an object lack human-centered context and informativeness, we extend them into descriptive situations based on holistic scene information, incorporating at least two reference objects, as illustrated in Fig.2.

```

system_prompt = ("You are an AI visual assistant tasked with expanding brief situational descriptions into 5 different detailed
situation descriptions with human-object interactions within a 3D scene. Initially, the situation involves only one reference object,
but your description should include at least two interacting objects. Exclude non-present objects. Each detailed description should
be less than 20 words. The response should be in the format with 'S' is the detailed description and 'O' is the reference objects.
Mention the directions (left, right, front, back) of all reference objects when standing. 'Interacting' should be an action conducted
while standing, with the interacted object in front. Don't assume 'interacting' to be 'sitting'.")

data_sit = {"windowsill.4": {"attributes": ["metal", "dark", "gray"], "location": "left"}, "plant.7": {"attributes": ["tall"],
"location": "left"}, "plant.8": {"location": "left, within arm reach"}, "beanbag.17": {"location": "left"}, "table.19":
{"attributes": ["wooden", "blue", "green", "rectangular", "low", "narrow"], "location": "front, within arm reach"}, "cushion.20":
{"location": "left, within arm reach"}, "cushion.21": {"attributes": ["tall", "wide"], "location": "left"}, "sofa.22": {"attributes":
["padded", "L-shaped", "orange", "pink", "wide"], "location": "below"}, "tv.24": {"attributes": ["black"], "location": "front, far
away"}... }

data_interact = {"sink.7": {"attributes": ["white"], "location": "front, within arm reach"}, "mirror.9": {"location": "front, within
arm reach"}, "toilet.13": {"attributes": ["seat down", "white", "tall", "wide"], "location": "left, within arm reach"}, "bucket.14":
{"location": "back, within arm reach"}, "trash can.16": {"location": "front, within arm reach"}... }

data_stand = { "kitchen counter.2": { "attributes": ["stone", "rectangular", "white", "low"], "location": "front, within arm
reach"}, "clutter.9": {"location": "front, within arm reach"}, "clutter.11": {"location": "front, within arm reach"}, "window.13":
{"attributes": ["glass", "white"], "location": "right"}, "garbage.16": {"attributes": ["cylindrical"], "location": "right"},
"doorframe.22": {"attributes": ["rectangular", "white"], "location": "left"}, "oven.24": {"attributes": ["black", "silver"],
"location": "right, within arm reach"}... }

example_sit = [
{"user": f "brief situation: sitting on sofa.22, object attributes: {data_sit}",
{"assistant": ""S": 'Sitting on the L-shaped sofa, watching TV far away.', 'O': 'sofa.22, tv.24' 'S': 'Sitting on sofa, chatting with a
person on the beanbag to my left.', 'O': 'sofa.22, beanbag.17' 'S': 'Sitting on sofa with a windowsill to the left.', 'O': 'sofa.22,
windowsill.4' 'S': 'Sitting on sofa with two plants to the left.', 'O': 'sofa.22, plant.7, plant.8' 'S': 'Sitting on the L-shaped sofa
with a wooden table in the front within arm reach.', 'O': 'sofa.22, table.19' 'S': 'Sitting on the L-shaped sofa with two cushions to
the left.', 'O': 'sofa.22, cushion.20, cushion.21'" }]

example_interact = [
{"user": f "brief situation: interacting with sink.7, object attributes: {data_interact}",
{"assistant": ""S": 'Washing hands at the sink in the front within arm reach, and a trash can to the left by my feet.', 'O': 'sink.7,
trash can.16' 'S': 'Cleaning the sink with a bucket behind me within arm reach.', 'O': 'sink.7, bucket.14' 'S': 'Washing my face at
the sink, while the toilet is to my left within arm's reach.', 'O': 'sink.7, toilet.13' 'S': 'Washing hands at the sink with a mirror in
the front within arm reach.', 'O': 'sink.7, mirror.9' 'S': 'Washing hands at the sink, with a small shelf to my left.', 'O': 'sink.7,
shelf.10'" }]

example_stand = [
{"user": f "brief situation: standing with kitchen counter.2 12 o'clock, object attributes: {data_stand}",
{"assistant": ""S": 'Baking in front of the kitchen counter, with an oven to my right within arm's reach.', 'O': 'kitchen
counter.2, oven.24' 'S': 'Cooking in front of the kitchen counter, with the doorframe to my left.', 'O': 'kitchen counter.2,
doorframe.22' 'S': 'Cooking in front of the kitchen counter, with a window to my right.', 'O': 'kitchen counter.2, window.13'
'S': 'Cooking in front of the kitchen counter, with a garbage to my right.', 'O': 'kitchen counter.2, garbage.16' 'S': 'Standing in
front of the kitchen counter, with two clutters also in the front within arm's reach.', 'O': 'kitchen counter.2, clutter.9, clutter.11'" }]

```

Figure 2: Prompt for descriptive situation generation.

## A.2 Long-Form Text Generation

Seven coauthors annotated each changed object in 3RScan [8] with situation-agnostic Reason, Description, Rearrangement Instruction, and Warning (Fig. 3). These were combined with object attributes, allocentric and egocentric relationships to generate situation-aware descriptions and instructions.

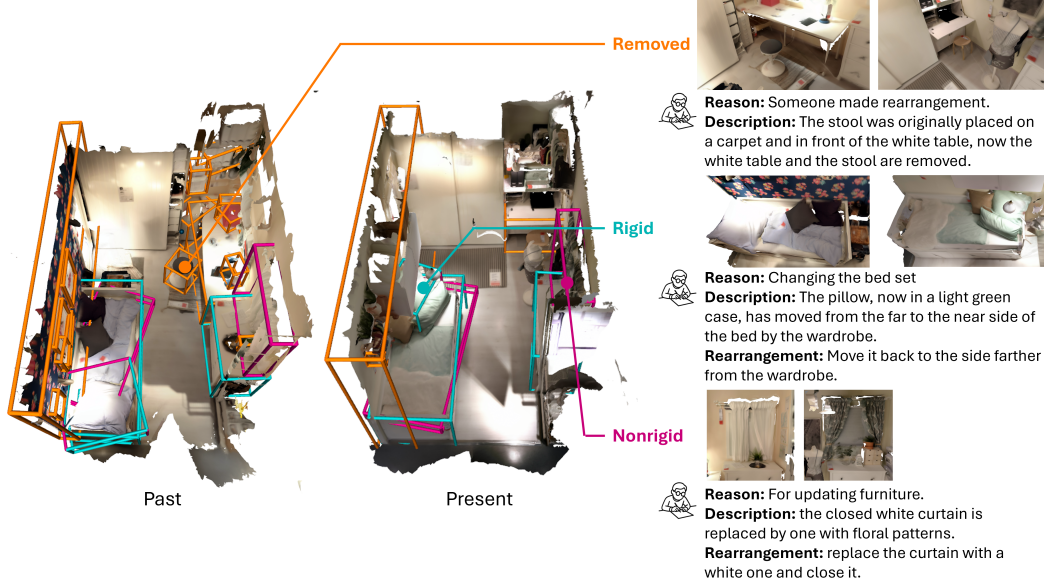


Figure 3: Human annotation for the original changes labeled in 3RScan [8] without situation-awareness. The original changes are categorized into **removed**, **rigid**, and **non-rigid** types.

`system_prompt` = ("You are an AI assistant tasked with generating captions of changes and instructions to rearrange changed objects in a 3D scene, based on the current location and orientation of the observer. This includes the vertical allocentric relationships among the objects, their horizontal locations (specified in degrees and distance) relative to the observer, and their attributes. Objects undergoing changes are classified into four categories: removed, added, rigid, and non-rigid. Always provide a caption ('C') that describes the change, including egocentric details, but exclude any rearrangement instructions ('R') for removed or added objects. To generate caption ('C'), rewrite 'Caption' to include at least one location with distance and clockwise direction: current ('location', 'allocentric') or original ('location\_old', 'allocentric\_old'), and the distance in 'return'. To generate 'C', don't use direction in 'return'. To generate rearrangement instruction ('R'), rewrite 'Instruction' to guide the user to reach the current 'location' of the changed object for the first step, then do 'return' to return the changed object, *i.e.* at least two steps for 'location' and 'return'. Mention the distance and direction of the movement ('location' and 'return'). And only generate 'C' for objects that have the label 'Caption'. When generating instructions, please always specify the direction and distance of the movement. Please rewrite the numbers (direction and distance) in 'Caption' and 'Instruction' with the provided ones ('location', 'location\_old', 'return'), adjust verbs (*e.g.*, push/pull) to reflect the observer's perspective. The output should be formatted as 'O' (object), 'T' (type of change), 'C' (description of change), and 'R' (numbered rearrangement actions, *e.g.*, '1., 2., 3,...')")

`data` = {"removed": {"storage\_22": {"location\_old": "4 o'clock, 0.4m", "Caption": "incomplete scan"}}, "rigid": {"table\_7": {"attributes": ["wooden", "rectangular", "white"], "location": "10 o'clock, 0.9m", "location\_old": "10 o'clock, 1.0m", "allocentric\_old": "monitor\_8 standing on table\_7, picture\_23 lying on table\_7", "allocentric": "monitor\_8 standing on table\_7", "Caption": "The table is against the wall, with a computer on top of it, and the window is to the right"}, "chair\_6": {"attributes": ["wide"], "location": "11 o'clock, 0.8m", "return": "2 o'clock, 0.5m", "location\_old": "12 o'clock, 1.0m", "Caption": "The chair was previously by the window, and now it is directly in front of the table.", "Instruction": "Move it one step right to the window"}}, "non\_rigid": {"curtain\_5": {"location": "11 o'clock, 1.5m", "allocentric": "curtain\_5 hanging on wall\_3", "Caption": "the change is not obvious"}}, "unchanged": {"rail\_33": {"location": "1 o'clock, 1.6m"}...}}

`example` = [{"user": "f" "brief situation: standing with chair\_34 9 o'clock, object attributes: {data}", {"assistant": "'O': 'storage\_22', 'T': 'removed', 'C': 'The partially scanned storage at your 4 o'clock, 0.6 meter away, may have been removed.' 'O': 'table\_7', 'T': 'rigid', 'C': 'The white table with a monitor on it at your 10 o'clock, 1.5 m away hasn't changed its position, but the picture on it has been removed.' 'O': 'chair\_6', 'T': 'rigid', 'C': 'The chair, which was at your 11 o'clock, 1.4 meters away by the window, has been moved 0.5 meter to the front of the table.', 'R': 1. Turn to your front-left and take two steps, bypassing the couch half a step away. 2. Pick up the chair in front of the table. 3. Move the chair one step to your right, placing it beneath the window.' 'O': 'curtain\_5', 'T': 'nonrigid', 'C': 'The curtain on your 1 o'clock, 1.8 meters away, remains hanging on the wall.'"}]

Figure 4: Prompt for generating situation-aware long-form texts, including change descriptions and rearrangement instructions.

### A.3 Query Generation for Long-Form Text

The queries for change description and rearrangement instruction are generated based on features that distinguish the object from others in the same category, as shown in Fig. 5. These features are filtered and refined by coauthors.

```
system_prompt = ("You are an AI assistant tasked with generating queries about changes to a specific object. Given the object's name and a set of its features, generate one query per feature. The tense indicates whether the provided information refers to the state before (past) or after (present) the change. Use the tense accordingly when generating queries, especially by referencing the spatial relation of the object (e.g., 'farthest object', 'nearest object', 'others', 'vertical_relationship'). 'others' also represents features of the object. 'num' represents the number of items in the same category within the scene. If 'num' equals 2, use the comparative form for the spatial location; if it is greater than 2, use the superlative form for the spatial location. Don't mention the instance ID of the object. Make the queries as short as possible to include only the necessary information. Please only ask for general changes, and don't ask about the specific change of the object.")
```

```
data_pre = {"object": "nightstand.8", "tense": "past", "num": 2,
"features": [{"nearest_objs": ["nearest to the curtain"]},
{"vertical_relationships": ["frame standing on nightstand", "lamp supported by nightstand"]}, {"farthest_objs": ["farthest to the wardrobe"]} ]}
```

```
data_post = {"object": "desk.5", "tense": "present", "num": 3,
"features": [{"nearest_objs": ["nearest to the wardrobe"]},
{"vertical_relationships": ["monitor standing on desk", "plant standing on desk"]} ]}
```

```
example_pre= [
"user": {data_pre},
"assistant": {"Query 1": "How has the nightstand that was nearer to the curtain been altered?", "Query 2": "Which updates have been made to the nightstand that had a frame and a lamp on it?", "Query 3": "Could you describe what modifications were applied to the nightstand that stood farther from the wardrobe?", "Query 4": "What changes have been made to the nightstand that stood farther from the wardrobe?", "Query 5": "How has the nightstand that was farther from the wardrobe been altered?", "Query 6": "What changes have been made to the nightstand that stood nearer to the curtain?", "Query 7": "What kind of changes were made to the nightstand set farther from the wardrobe?", "Query 8": "How has the nightstand that was nearer to the curtain been altered?", "Query 9": "Please explain what has been adjusted on the nightstand situated farther from the wardrobe.", "Query 10": "What revisions have taken place regarding the nightstand that was at a distance from the wardrobe?" } }
```

```
example_post= [
"user": {data_post},
"assistant": {"Query 1": "How has the desk that is nearest to the wardrobe been altered?", "Query 2": "What changes have been made to the desk that is closest to the wardrobe?", "Query 3": "Which updates have been made to the desk that has a monitor and a plant on it?", "Query 4": "Could you describe what modifications were applied to the desk with a monitor and a plant on it?", "Query 5": "How has the desk that is positioned nearest to the wardrobe been altered?", "Query 6": "What modifications have been applied to the desk situated nearest to the wardrobe?", "Query 7": "What kind of changes were made to the desk that is closest to the wardrobe?", "Query 8": "How has the desk with a monitor and a plant on it been altered?", "Query 9": "Please explain what has been adjusted on the nightstand situated farther from the wardrobe.", "Query 10": "What revisions have taken place regarding the desk that is closest to the wardrobe?" } }
```

Figure 5: Prompt for generating queries for long-form texts, with an example for change description.



#### A.4 QA Generation

The QA pairs are generated using object attributes, as well as egocentric and allocentric relationships, following O-CoT [3] as shown in Fig. 6. Each pair may include the object’s label, index, and QA type to retrieve the correct answer from the original data. Figure 7 illustrates examples of QA types.

`system_prompt` = (“You are an AI visual assistant tasked with generating question and answer pairs based on changes observed in a sequence of scene images. The scenes detail the journey along a familiar route, highlighting shifts in object positioning and attributes. Your questions should cover the following areas:  
Warning: Query if there is any changed object that obstructs the familiar route to a target object. If an object has the attribute ‘Warning’ means it becomes an emerged obstacle towards the target object in the list. Only mention one target object in the question. Egocentric Distance Old/ Egocentric Distance (‘How far ...’): Calculate the distance from the observer to the current or original location of objects. Prioritize the ‘egocentric distance old’ if the change exists. Allocentric Displacement (‘How far ...’): ask about ‘move\_distance’ of a specific object. Egocentric Direction Old/ Egocentric Direction (‘In which direction ...’): Determine the current or original orientation of objects in relation to the observer. Prioritize the ‘egocentric direction old’ if the change exists. Allocentric Relationship (‘Where’): Examine the old or current vertical spatial relationships between objects. Counting: Count objects of a specific type in a direction to the observer (front, left, behind, right). Existence: Note the addition or removal of specific objects. Attribute: Ask about a specific aspect of an object, focusing on its status, color, and material. Questions start with like ‘What is the status/ color/ material?’. Affordance: Check for objects serving specific purposes in the observer’s immediate vicinity. For each scenario, generate 15 questions and answer pairs addressing these topics to effectively map the changes in the scene. Don’t ask anything about the wall, the ceiling, or the floor. Don’t answer the direction and distance together. Don’t mention numbers in the question. ‘Where’ is only for an egocentric relationship. Each answer should be a maximum of 5 words. Exclude non-present objects. Don’t ask questions that cannot be answered. Don’t ask for the direction of the movement. Please don’t confuse shape with size. The output is in the format with ‘Q’ for the question, ‘A’ for the answer, ‘O’ for the reference object, and ‘Type’ for the type of question and answer pairs.” )

`data` = {“rigid”: {“chair\_39”: {“location”: “11 o’clock, 0.4m”, “move\_distance”: “1.6m”, “location\_old”: “10 o’clock, 1.7m”, “Warning”: [“bed\_3”, “cabinet\_6”, “bag\_24”, “bag\_38”, “blanket\_40”, “laptop\_41”, “roll\_42”]}, “table\_18”: {“material”: [“wooden”], “color”: [“white”], “shape”: [“rectangular”], “state”: [“messy”], “location”: “5’clock, 0.1m”, “location\_old”: “5 o’clock, 0.0m”}}, “unchanged”: {“bed\_3”: {“state”: [“messy”], “color”: [“gray”], “size”: [“low”, “narrow”], “location”: “11 o’clock”, 1.1m}, “cabinet\_6”: {“size”: [“big”], “location”: “12 o’clock, 1.9m”}, “door\_14”: {“state”: [“closed”], “shape”: [“flat”], “color”: [“white”], “size”: [“tall”], “location”: “2 o’clock, 3.9m”, “allocentric”: “hanging on wall\_13”}, “clothes\_15”: {“color”: [“beige”, “black”], “location”: “2 o’clock, 3.8m”, “allocentric”: “hanging on door\_14”}, “chair\_19”: {“location”: “6 o’clock, 0.3m”}, “clothes\_dryer\_20”: {“location”: “9 o’clock, 1.9m”}, “window\_22”: {“material”: [“glass”], “shape”: [“rectangular”], “state”: [“half open/closed”], “location”: “9 o’clock, 1.6m”, “allocentric”: “attached to wall\_4, attached to wall\_2”}, “basket\_25”: {“size”: [“big”], “location”: “8 o’clock, 1.4m”}, “clothes\_dryer\_37”: {“location”: “2 o’clock, 3.1m”}, “blanket\_40”: {“location”: “11 o’clock, 1.8m”, “allocentric”: “lying on bed\_3”}, “laptop\_41”: {“color”: [“gray”], “location”: “12 o’clock, 2.4m”, “allocentric”: “standing on bed\_3”}...}

`example`= [  
{“user”: {data},  
{“assistant”: (““Q: ‘How far was the chair, which was between the clothes dryer and the bed, moved?’, ‘A: ‘1.6 m’, ‘O: ‘chair\_39’, ‘Type’: ‘Allocentric Displacement’” “Q: ‘Are there any changed objects on my familiar route to the bed?’, ‘A: ‘A chair’, ‘O: ‘bed\_3, chair\_39’, ‘Type’: ‘Warning’” “Q: ‘What is the status of the white wooden table?’, ‘A: ‘Messy’, ‘O: ‘table\_38’, ‘Type’: ‘Attribute’” “Q: ‘How far was the clothes dryer to my left relative to me?’, ‘A: ‘1.9 m’, ‘O: ‘clothes dryer\_20’, ‘Type’: ‘Egocentric Distance Old’” “Q: ‘How many chairs are there behind me?’, ‘A: ‘One’, ‘O: ‘chair\_19’, ‘Type’: ‘Counting’” “Q: ‘Is there something to hang clothes on in this room?’, ‘A: ‘Two clothes dryers’, ‘O: ‘clothes dryer\_20, clothes dryer\_37’, ‘Type’: ‘Affordance’” “Q: ‘Which direction was the changed chair relative to me?’, ‘A: ‘10 o’clock’, ‘O: ‘chair\_39’, ‘Type’: ‘Egocentric Direction Old’” “Q: ‘Is there any sofa in the room?’, ‘A: ‘No’, ‘O: ‘None’, ‘Type’: ‘Existence’” “Q: ‘Is there anything to keep warm while sleeping?’, ‘A: ‘A blanket’, ‘O: ‘blanket\_40’, ‘Type’: ‘Affordance’” “Q: ‘Where are the beige and black clothes?’, ‘A: ‘Hanging on the door’, ‘O: ‘clothes\_15, door\_14’, ‘Type’: ‘Allocentric Relationship’” “Q: ‘Where is the laptop?’, ‘A: ‘Standing on the bed’, ‘O: ‘laptop\_41, bed\_3’, ‘Type’: ‘Allocentric Relationship’” “Q: ‘How far is the basket from me?’, ‘A: ‘1.4 m’, ‘O: ‘basket\_25’, ‘Type’: ‘Egocentric Distance’” “Q: ‘What is the status of the window?’, ‘A: ‘Half open’, ‘O: ‘window\_22’, ‘Type’: ‘Attribute’” “Q: ‘Which direction is the changed chair relative to me?’, ‘A: ‘11 o’clock’, ‘O: ‘chair\_39’, ‘Type’: ‘Egocentric Direction’” “Q: ‘How far is the chair in front of me?’, ‘A: ‘40 cm’, ‘O: ‘chair\_39’, ‘Type’: ‘Egocentric Distance’”) } ]

Figure 6: Prompt for QA generation.

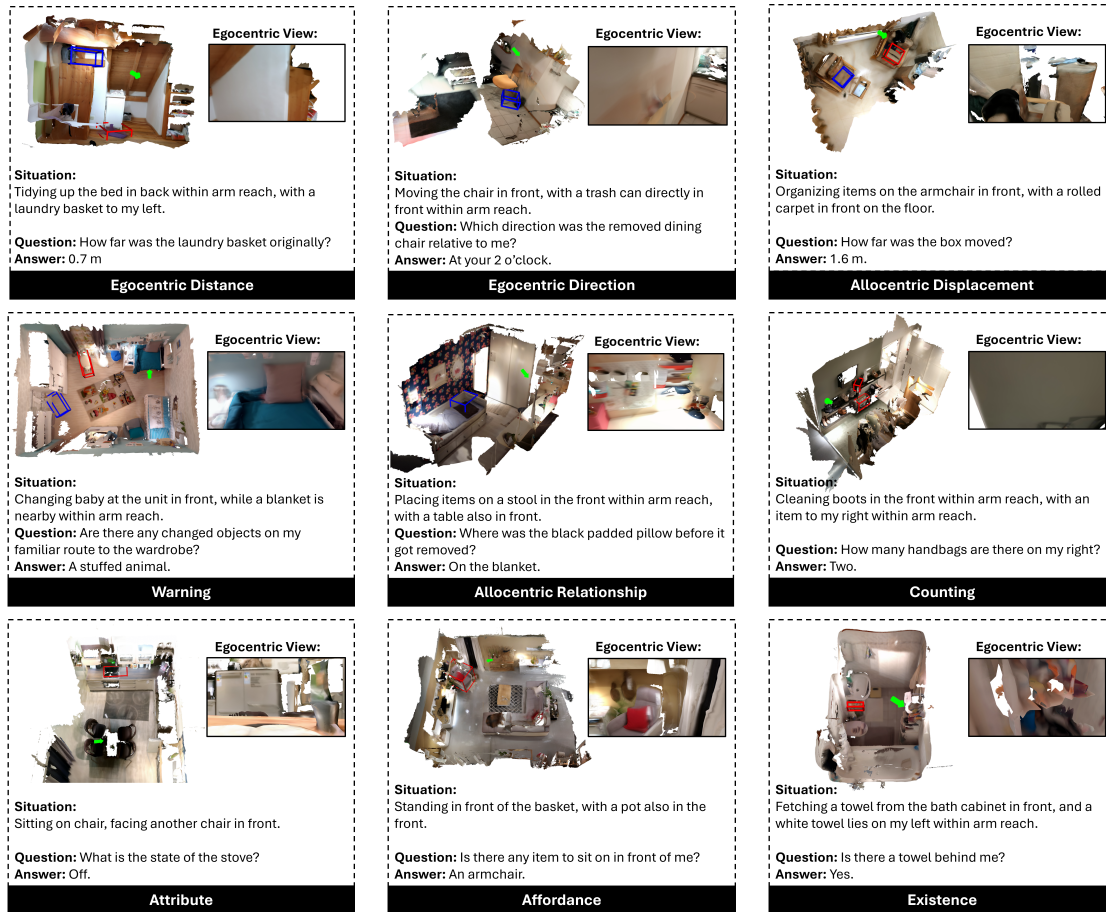


Figure 7: Examples of the nine QA types. **Blue** indicates the previous position, and **red** indicates the present position.

## B Dataset Statistics

Table 1. Statistics of the Situat3DChange dataset on scenes, situations, and changes.

|            | General    |               | Number of Situations |      |          | Number of Changes |         |           |
|------------|------------|---------------|----------------------|------|----------|-------------------|---------|-----------|
|            | Scan Pairs | Total Objects | Stand                | Sit  | Interact | Rigid             | Removed | Non-Rigid |
| Training   | 793        | 22274         | 5981                 | 1550 | 3107     | 2543              | 446     | 504       |
| Validation | 110        | 3532          | 882                  | 192  | 433      | 390               | 96      | 54        |

Table 2. Distribution of QA types related to scene changes.

| Allo. Dis. | Warning | Ego. Dir. | Allo. Rel. | Attribute | Existence | Ego. Dis. | Affordance | Counting |
|------------|---------|-----------|------------|-----------|-----------|-----------|------------|----------|
| 100.00%    | 100.00% | 53.41%    | 40.03%     | 15.88%    | 23.79%    | 21.66%    | 8.91%      | 6.91%    |

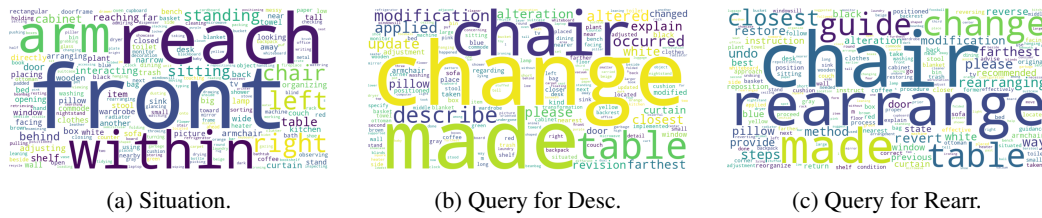


Figure 8: Word clouds of situations, change description queries, and rearrangement instructions.

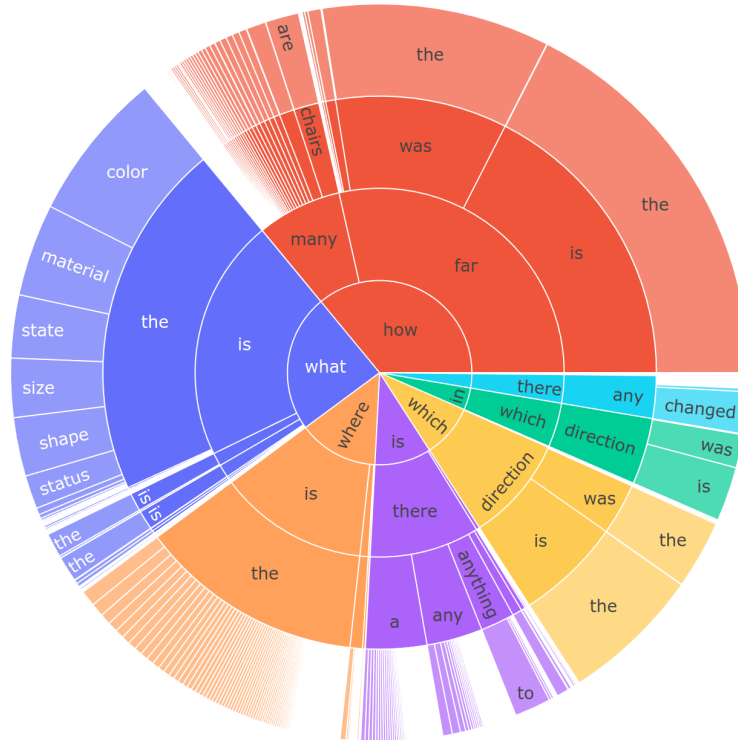


Figure 9: Hierarchical distribution of questions in Situat3DChange.

## C MLLM Paradigms for Situat3DChange

The paradigms of the baselines are shown in Fig.10, with the tokenizer for system prompts and the decoder for model responses omitted for simplicity. On the right side of Fig.10 (b) is our SCReasoner, which differs from LEO only in the fusion comparison module. This module compares similar tokens from the previous and current scenes, leveraging Mamba’s [2] selective token processing and a star operation [5] for fusion. It injects half of the scene tokens into the LLM as LEO does, focusing on the most relevant differences between the scene pair. The hyperparameters for fine-tuning are listed in Tab. 3.

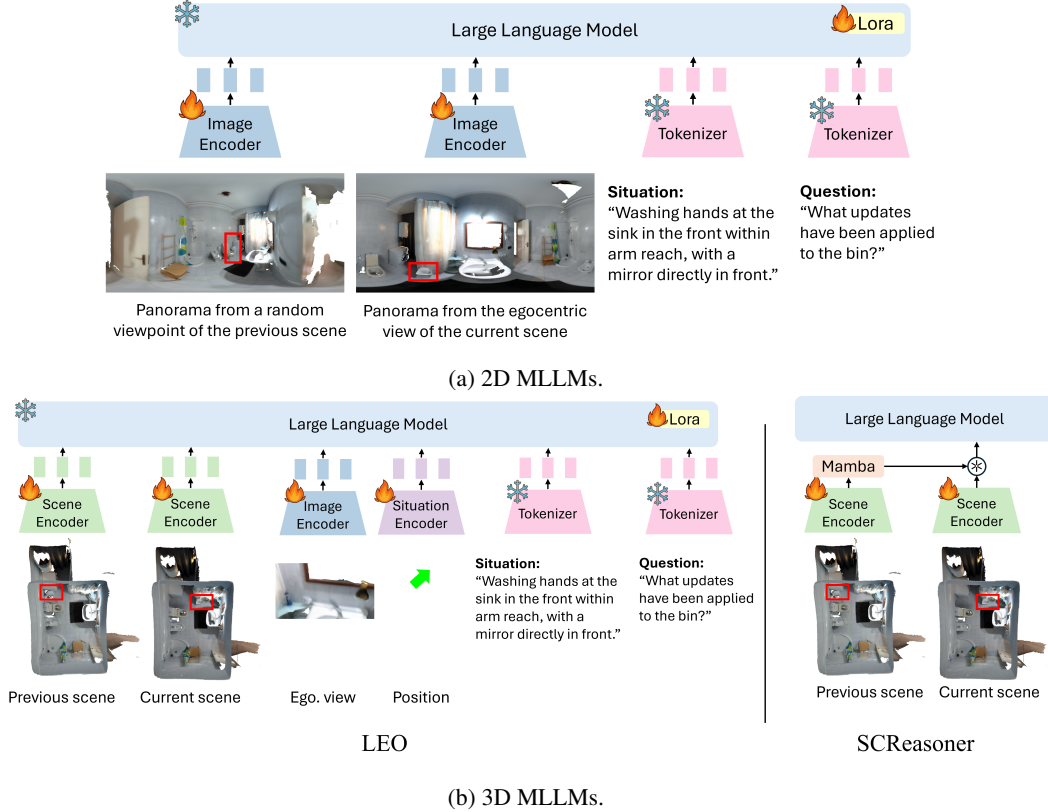


Figure 10: Baseline paradigms. Modules with the same color share weights.

Table 3. Hyperparameters for fine-tuned models.

| Hyperparameter     | InternVL2 [1]      | LEO [3]            | SCReasoner         |
|--------------------|--------------------|--------------------|--------------------|
| Optimizer          | AdamW              | AdamW              | AdamW              |
| Weight decay       | 0.05               | 0.05               | 0.05               |
| Betas              | [0.9, 0.999]       | [0.9, 0.999]       | [0.9, 0.999]       |
| Learning rate      | $4 \times 10^{-5}$ | $3 \times 10^{-5}$ | $3 \times 10^{-5}$ |
| Warmup steps       | 2852               | 400                | 400                |
| Number of workers  | 4                  | 4                  | 4                  |
| Parallel strategy  | DDP                | DDP                | DDP                |
| Type of GPUs       | A100               | A100               | A100               |
| Memory             | 40 GB              | 40 GB              | 40 GB              |
| Number of GPUs     | 4                  | 4                  | 4                  |
| Batch size per GPU | 2                  | 2                  | 2                  |
| Training precision | bfloat16           | bfloat16           | bfloat16           |
| Epochs             | 5                  | 5                  | 5                  |

## D Evaluation Details

### D.1 Prompts for GPT Scoring

We craft fine-grained prompts to evaluate open-ended responses from MLLMs: general QA in Fig. 11, direction-related QA in Fig. 12, and long-form text in Fig. 4. For evaluation, we use GPT-4o-mini [7] (timestamp 2024-07-18), which has been shown to outperform GPT-3.5 in both accuracy and cost-efficiency.

Score open-ended answers from 1 to 5 based on accuracy to the ground truth.  
Score 2-4: Reflect partial correctness or minor errors.  
Criteria:  
Affordance: Question: Is there any furniture to rest feet on nearby? Ground Truth: Yes. Example Response: Yes, there is an ottoman nearby. Score: 5 (Correct match). Attribute: Question: What is the color of the ottoman? Ground Truth: Blue, red, brown. Example Response: The ottoman is brown. Score: 3 (Partial match). Existence: Question: Is there a chair on my left? Ground Truth: Yes. Example Response: Yes, there is a chair on the left. Score: 5 (Correct match). Counting: Question: How many tables are in the room? Ground Truth: Three Examples. Response: Two. Score: 1 (Significant discrepancy). Warning: Question: Are there any changed objects on my familiar route to the door? Ground Truth: Yes, a chair. Example Response: Yes, there is a table on the way to the door. Score: 2 (Major incorrect). Allocentric Relationship: Question: Where is the kettle? Ground Truth: On the kitchen cabinet. Example Response: The kettle is on the kitchen counter. Score: 4 (Approximate match).  
Output only the score.

Figure 11: Prompt for LLM-assisted scoring of general QA.

Score open-ended answers from 1 to 5 based on accuracy to the ground truth.  
Score 2-4: Reflect partial correctness or minor errors.  
Mapping of proximity direction and clock face: front (from 11 to 1 o'clock), left (from 8 to 10 o'clock), right (from 2 to 4 o'clock), back (from 5 to 7 o'clock).  
Criteria:  
Score 5: If the difference is less than or equal to 1 o'clock on the clock face, *e.g.*, GT: '11 o'clock', Response: '10 o'clock'.  
Score 4: If the response is in the correct proximity direction, *e.g.*, GT: '6 o'clock'(back), Response: 'Back'. Score 3: If the response is adjacent to the correct direction, *e.g.*, GT: '11 o'clock'(front left), Response: 'Left'. Score 2: If the response has a significant directional error but is not completely opposite, *e.g.*, GT: '3 o'clock'(right), Response: 'Back'.  
Score 1: If the response is in the opposite proximity direction to the ground truth, *e.g.*, GT: '9 o'clock'(left), Response: '4 o'clock'(right).  
Output only the score.

Figure 12: Prompt for LLM-assisted scoring of egocentric direction QA.

You are an intelligent evaluator tasked with assessing the correctness and semantic similarity of model-generated answers to question-answering pairs. Your goal is to compare the predicted answer with the reference (correct) answer and assign a score based on how well they align in meaning. Use the following scoring rubric:  
Score 5: Completely correct or semantically equivalent.  
Score 4: Key information is correct, with minor inaccuracies or omissions.  
Score 3: Some relevant information, but lacks sufficient correctness or completeness.  
Score 2: Mostly incorrect, but shows some relevance to the question.  
Score 1: Completely incorrect or nonsensical.  
Your response must be a single integer from 1 to 5, with no additional text or explanation.

Figure 13: Prompt used for LLM-assisted scoring of long-form texts, including change descriptions and rearrangement instructions.

## D.2 Alignment between Human and GPT Evaluation

To validate the validity of the GPT-based evaluation results, We recruited and acknowledge four human evaluators who are not involved in this project. We selected SCReasoner, LEO, and one-shot InternVL for human evaluation. SCReasoner and LEO are compared to demonstrate our improvements, while one-shot InternVL represents an open-ended, training-free LLM. For each model, we randomly selected 40 samples for each QA type (excluding those related to distance) and each long-form task, resulting in the same 360 samples per model. As shown in the table, SCReasoner consistently outperforms LEO on the 360 sampled instances, with the performance gap further amplified when evaluated by human scores compared to GPT scores.

Table 4. Human and GPT evaluation results.

| Model      | Description |       | Rearrangement |       | QA   |       |
|------------|-------------|-------|---------------|-------|------|-------|
|            | GPT         | Human | GPT           | Human | GPT  | Human |
| InternVL   | 4.0         | 6.5   | 3.0           | 7.5   | 33.1 | 36.4  |
| LEO        | 11.5        | 14.5  | 22.5          | 19.5  | 43.3 | 45.6  |
| SCReasoner | 14.0        | 20.5  | 26.0          | 31.5  | 48.3 | 50.9  |

Following OpenEQA [6], we computed the Spearman correlation between human scores and GPT-generated scores. The GPT scores show a strong correlation with human evaluation ( $\rho > 0.6$ ), indicating that GPT-based evaluation aligns well with human judgment.

Table 5. Spearman correlation between GPT and human evaluations.

|                | Description | Rearrangement | QA   | Average |
|----------------|-------------|---------------|------|---------|
| Spearman Corr. | 0.75        | 0.70          | 0.94 | 0.80    |



## E Ablation Studies

### E.1 Consistent Improvements with Run-to-Run Validation

In the main text, we followed LEO’s setting by fixing the random seed. To better analyze the errors and demonstrate the performance gain of SCReasoner over LEO, we conducted three additional runs with different random seeds, resulting in four runs in total. The mean scores and standard deviations for each task are reported in Tab. 6, showing that SCReasoner consistently outperforms LEO.

Table 6. Results of LEO and SCReasoner (mean  $\pm$  std over 5 seeds).

|            | QA                 | Description        | Rearrangement      |
|------------|--------------------|--------------------|--------------------|
| LEO        | 52.760 $\pm$ 0.511 | 12.733 $\pm$ 0.117 | 30.165 $\pm$ 0.410 |
| SCReasoner | 53.420 $\pm$ 0.232 | 13.629 $\pm$ 0.287 | 30.753 $\pm$ 0.216 |

### E.2 Panorama vs. Multiview Input for 2D MLLMs

Since 2D MLLMs are rarely trained on panoramas and are only weakly exposed to multi-image understanding, we stitched the four surrounding views with the front view placed at the center to form a single surround-view image as input to the one-shot InternVL model, in order to assess the plausibility of using panoramas. As shown in Table 7, the performance difference between panorama and stitched multi-view inputs is negligible.

Table 7. Comparison of panorama and multi-view inputs for 2D MLLM performance.

| Input      | QA   | Description | Rearrangement |
|------------|------|-------------|---------------|
| panorama   | 35.7 | 3.8         | 3.7           |
| multi-view | 35.5 | 3.7         | 3.5           |

### E.3 Understanding the Underperformance of 2D MLLMs

The short QA results in Tab. ?? can be regarded as a decomposition of the comprehensive understanding of dynamic situations and environments. We observe that 2D MLLM performs better than 3D MLLMs in terms of egocentric information with the similar model size, while underperforming in allocentric reasoning and object property identification. This indicates that although panoramic views can effectively reconstruct visible areas and convey orientation information [10], they still suffer from occlusion issues and limited field of view constraints.

To further analyze this limitation, we report results in Tab. 8 for long-form tasks using two 2D MLLMs with one-shot CoT prompting, compared against one-shot results without CoT. We prompt the models to separately analyze the two panoramic views and then compare them based on the current situation, in order to better capture allocentric information. This strategy yields notable performance gains for both models, except that Janus remains on par in the rearrangement task, underscoring the importance of allocentric reasoning. Nevertheless, their performance still lags behind parameter-efficient fine-tuned counterparts.

Table 8. Effect of chain-of-thought (CoT) prompting on description and rearrangement tasks.

| Model    | Description (wo CoT) | Description (with CoT) | Rearrangement (wo CoT) | Rearrangement (with CoT) |
|----------|----------------------|------------------------|------------------------|--------------------------|
| InternVL | 3.8                  | 5.3                    | 3.7                    | 4.9                      |
| Janus    | 2.7                  | 3.6                    | 4.7                    | 4.3                      |

## References

- [1] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024. 8
- [2] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *COLM*, 2024. 8
- [3] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3D world. In *ICML*, 2024. 5, 8
- [4] Xiongkun Linghu, Jiangyong Huang, Xuesong Niu, Xiaojian (Shawn) Ma, Baoxiong Jia, and Siyuan Huang. Multi-modal situated reasoning in 3D scenes. In *NeurIPS*, 2024. 1
- [5] Xu Ma, Xiyang Dai, Yue Bai, Yizhou Wang, and Yun Fu. Rewrite the stars. In *CVPR*, 2024. 8
- [6] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul McVay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent-Pierre Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Alexander Sax, and Aravind Rajeswaran. OpenEQA: Embodied question answering in the era of foundation models. In *CVPR*, 2024. 10
- [7] OpenAI. Gpt-4o and gpt-4o-mini. OpenAI Model Card, 2024. <https://platform.openai.com/docs/models/gpt-4o>. 9
- [8] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. RIO: 3D object instance re-localization in changing indoor environments. In *ICCV*, 2019. 1, 3
- [9] Johanna Wald, Helisa Dharmo, Nassir Navab, and Federico Tombari. Learning 3D semantic scene graphs from 3D indoor reconstructions. In *CVPR*, 2020. 1
- [10] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. PanoContext: A whole-room 3D context model for panoramic scene understanding. In *ECCV*, 2014. 11