

# Appendix

## Table of Contents

---

<b>A: Extended Related Work</b>	11
<b>B: Constructing Receptacle-Aware 3D Scene Graph</b>	11
B.1: Scene Graph Initialization	11
B.2: Calculating Receptacle Regions	12
B.2.1: Initial Receptacles	12
B.2.2: Refined Receptacles	13
B.2.3: Merging Receptacles	13
<b>C: Generating Outcome-based Tasks</b>	13
C.1: Details of MANITASKOT-200 Dataset	13
C.1.1: Dataset Collection Process	13
C.1.2: More Examples of MANITASKOT-200	14
C.2: VLM-based Voting Mechanism	15
<b>D: Experiment Details</b>	15
D.1: Details of ManiTaskGen-RAS-40K	15
D.1.1: Scene Data Preprocessing	15
D.1.2: Generating Tasks of Controlled Complexity and Length	17
D.2: Examples of Complete Benchmark Episodes	19
D.2.1: Successful Episode	19
D.2.2: Failed Episode	52

## A Extended Related Work

**Improvement of VLM-based Agents.** Numerous research has explored various strategies to improve LLM/VLM-based digital agents [48, 47, 49, 27, 50] and embodied agents [51, 52, 15, 7, 53]. One important line of work employs collected demonstrations for supervised fine-tuning (SFT) [15, 48, 54, 55]. Given sufficient data, SFT ensures the resulting model achieves comparable performance with the source demonstrations. However, demonstration collection is labor-intensive and expensive, making it challenging to scale across diverse tasks and scenarios. Another prominent thread of methods leverages Reinforcement Learning (RL) [47, 56, 26, 57] to tune agent policies, through learning from reward models based on human feedbacks [58, 56], task-specific handcrafted reward functions [59, 47], or reward generators [51, 26]. To demonstrate the application of ManiTaskGen tasks for agent improvement, we designed and implemented a simple inference-time RL method inspired by Reflexion [26] and ReAct [27]. The method leverages the generated tasks and evaluations for self-reflection, and experimental results demonstrate its effectiveness. Future work involves extending ManiTaskGen’s compatibility with more advanced RL-based improvement methods [51, 52]. Overall, ManiTaskGen produces abundant resources for agent improvement, including task instructions as well as final and sub-step evaluations, thereby empowering various approaches to tune VLM-based agents.

## B Constructing Receptacle-Aware 3D Scene Graph

As described in Sec. 3.2, the process of constructing the 3D scene graph mainly consists of 2 stages: (1) Initializing the scene graph based on mutual spatial relationships between objects; (2) Calculating receptacle regions anchored by objects. We provide further details on each part below.

### B.1 Scene Graph Initialization

First, based on each object’s pose and the size of its bounding box, we infer the spatial relationships between objects along the vertical axis. These spatial relationships are used to initialize the scene graph. In the scene graph, each node represents an object, and a child node indicates that the corresponding object is placed on the upper surface of its parent. For scenes that provide object mesh models (e.g., ReplicaCAD and AI2THOR), we additionally extract internal surfaces suitable for object placement and record them in nodes’ attributes. Consequently, the scene graph is extended such that a child node can be placed on either an internal or external surface of its parent node. To facilitate subsequent identification and indexing of receptacles within the scene, we also define the following attributes for each node (i.e., each object) during the scene graph initialization stage.

**Ground Objects and Surface Objects.** We divide all objects in a scene into ground objects and surface objects, where ground objects are those resting directly on the ground, while surface objects are those supported by other objects. We assume the robot can only manipulate surface objects.

**Walkable Spaces.** For each ground object, we define the space unobstructed in the vertical direction of its bounding box as its walkable space. These walkable spaces serve as navigation targets for the robot when traversing the scene. For surface objects placed on ground objects, the corresponding ground object’s walkable space is also used as the robot’s navigation target when approaching those surface objects. In our implementation, we adopt a threshold of 50 cm to determine whether the unobstructed space in a given direction is sufficient to be considered walkable. A single ground object may have multiple walkable spaces, corresponding to different edges of its bounding box.

**Headings of Objects.** To support the generation of direction-related tasks (e.g., placing object A to the left of object B), we also define the heading of each object. For ground objects, we assume that their heading is determined by the robot’s standing orientation when navigating to the area around the object (decided by the walkable spaces)—specifically, the direction the robot is facing is taken as the *front* (heading) of the ground object, as shown in Fig.8 (a). If a ground object has multiple walkable spaces, its heading is not fixed and depends on the robot’s actual standing orientation. For surface objects, we assume their heading always aligns with that of the supporting ground object. Therefore, for surface objects whose bounding boxes are not parallel to those of their supporting ground objects, we still use the ground object’s heading as the sole reference when reasoning about their directional relationships.



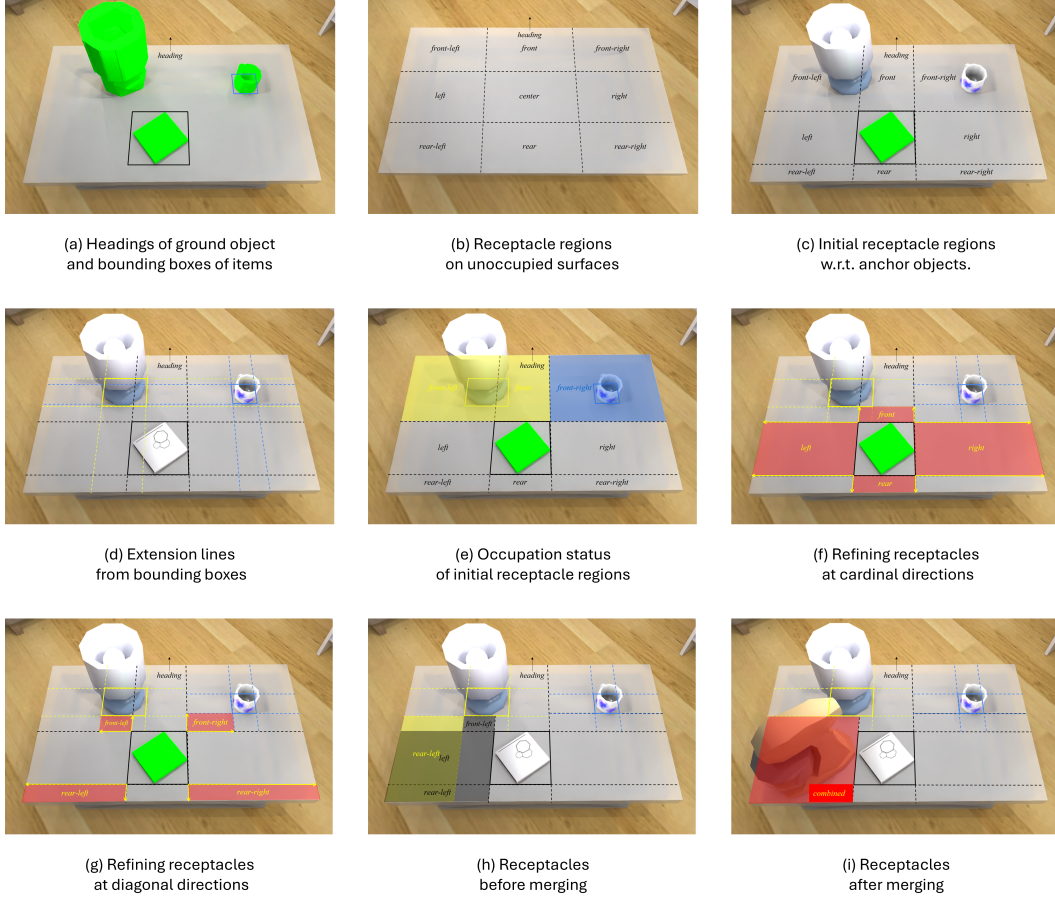


Figure 8: Visualization of Rules Involved in Scene Graph Construction.

## 460 B.2 Calculating Receptacle Regions

461 We construct accurate receptacle region retrieval information in three steps. First, we compute an  
 462 initial set of rough receptacles based on empty surfaces or the spatial relationships between existing  
 463 objects and the supporting surfaces (Sec. B.2.1). Next, we refine these receptacles by considering the  
 464 relative positions of objects on the same surface (Sec. B.2.2). Finally, we introduce rules for merging  
 465 adjacent or overlapping receptacles (Sec. B.2.3).

### 466 B.2.1 Initial Receptacles

467 Initial receptacles is computed mainly to better determine the relative positioning of objects.

468 For unoccupied surfaces, we divide the area into a 3x3 grid consisting of nine identical rectangular  
 469 regions, as visualized in Fig. 8 (b). The interior rectangle is defined as the "center" region. For the  
 470 surrounding eight regions, we designate their directions based on the surface's heading: the direction  
 471 aligned with the heading is labeled "front," with the remaining regions proceeding counterclockwise  
 472 as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." These receptacle  
 473 regions are directly indexed by surfaces in the scene graph. For occupied surfaces, we use a different  
 474 approach while maintaining the same directional terminology, as shown in Fig. 8 (c) and (d). We treat  
 475 each surface object as an anchor object to index the receptacle regions around it. Specifically, we  
 476 extend the vertices and edges of each object's bounding box outward until they reach the platform's  
 477 boundaries. The regions swept by the extended edges define the object's initial receptacles in four  
 478 cardinal directions (front, rear, left, right), while the paths traced by the extended vertices determine  
 479 the initial receptacles in the four diagonal directions (front-left, front-right, rear-left, rear-right).

480 With these initial receptacles for each anchor object, we then determine the relative positioning of  
481 other objects on the same surface by identifying which of our calculated initial receptacles they  
482 overlap with. Fig. 8 (e) illustrates this process. The relative positioning and occupation information  
483 will also be recorded in the scene graph.

### 484 B.2.2 Refined Receptacles

485 We proceed to refine the initial receptacles based on accurate object positions and bounding boxes,  
486 and record their properties into the scene graph.

487 For cardinal directions, we project the edge of an object’s bounding box along its perpendicular  
488 direction until it intersects another object’s bounding box or the platform boundary. The region swept  
489 during this projection forms the refined receptacle in this cardinal direction, as shown in Fig. 8 (f).

490 For diagonal directions, we project the vertices of the object’s bounding box along two corresponding  
491 cardinal directions until they encounter the extension line of another object’s bounding box or  
492 the platform boundary. The rectangle formed by these two projection paths represents the refined  
493 receptacle in this diagonal direction, as illustrated in Fig. 8 (g).

494 Thus, these refined receptacles are treated as final ones indexed by anchor objects. Together with the  
495 receptacle regions indexed directly by empty surfaces, they establish a systematic representation for  
496 efficient and retrieval of the free spaces on surfaces.

### 497 B.2.3 Merging Receptacles

498 Merging receptacles is applied in task generation and during interaction for object placement, to  
499 support retrieving multiple refined receptacles.

500 When generating process-based tasks, for any surface object in the scene, we traverse all receptacle  
501 regions in the scene to determine whether there is sufficient space to place the object. If not,  
502 for multiple receptacle regions belonging to the same surface, we attempt to merge up to four  
503 adjacent or overlapping receptacle regions. If the merged receptacle region provides enough space to  
504 accommodate the object, we then generate the corresponding task. Fig. 8 (h) shows an example of  
505 merging four receptacles when generating the task of placing a large kitchen utensil in the scene.

506 During interaction between the agent and the environment, the agent may select any number of  
507 receptacles on a surface, regardless of whether they are indexed by the same anchor object. The  
508 system will then attempt to merge them into a larger receptacle region if they are connected or  
509 overlapped, and to assess if the resulted one has enough space to fit the object, as shown in Fig. 8 (i).

## 510 C Generating Outcome-based Tasks

### 511 C.1 Details of MANITASKOT-200 Dataset

#### 512 C.1.1 Dataset Collection Process

513 The MANITASKOT-200 dataset contains 200 templates for outcome-based tasks, processed based  
514 on the human instructions collected from Amazon Mechanical Turk (AMT) [60]. We first rendered  
515 scene images from real-world and simulated datasets, including Habitat [23], ScanNet [61], and  
516 SUN-RGBD [25]. Next, we launched an instruction collection job in AMT [60] to gather natural  
517 language instructions from human annotators specific for the provided images. The detailed job  
518 description is shown below:

519 **General Description:** Imagine a powerful mobile robot that can move any object  
520 except immovable ground furniture. Describe in natural language the tasks it can  
521 perform, especially high-level complex tasks.

522 **Detailed Instruction:** Imagine we have a powerful mobile robot that can move  
523 and place any non-ground object in a room. Your task is to describe high-level,  
524 complex tasks the robot can perform in this room using natural language. Pictures  
525 of the room and its layout are provided here (link). First, please review layout.png,  
526 [0][ROOM].png, and then all the other images to understand the room layout and

object reference information. Make sure your tasks are reasonable and closely follow the provided images.

**Key Points:** (1) For immovable ground objects in the room, when referring to these objects, strictly use the numbered labels listed below: [1][DESK], [2][SHELVING], etc. (2) When referring to other small objects, you can describe them freely, but make sure to enclose the object names in square brackets, like this: [book], [plastic bottle], [small robotic arm]. (3) Please use your imagination as much as possible and give complex, high-level, and imaginative instructions.

**Examples of tasks we want:**

- (1) Help me clear the top of [1][DESK].
- (2) Organize all [soda cans] on the top layer of [2][SHELVING] to line them up in a row.
- (3) Tidy up the [2][SHELVING] to make it aesthetically pleasing.
- (4) The [7][DESK] is too messy. Organize the objects on top of it to make it neat.
- (5) Take out the [cans] from under the [paper box] on the left side of [3][DESK] and arrange them into an equilateral triangle.

We view these examples as positive ones because they contain complex, high-level instructions.

**Examples of tasks we do NOT want:**

- (1) Move the [paperbox] under [1][DESK] to [8][DESK]
  - (2) Pick up the [tiny robotic arm] from [3][DESK] and place it on the [5][CHAIR].
- We view these examples as negative ones because these tasks are too simple or the instructions are too detailed.

Based on the collected human instructions, we manually filtered out low-quality and repeated ones and removed specific object names in each instruction to create the final templates. In the end, we organized a set of 200 task templates, named as MANITASKOT-200.

### C.1.2 More Examples of MANITASKOT-200

We provide more templates from MANITASKOT-200 as follows:

- "Align all [SUB-PLATFORM-OBJECTS00] on [PLATFORM0] symmetrically."
- "Create a "staircase" on [PLATFORM0] by arranging the objects to resemble steps."
- "Create a pattern with the [SUB-PLATFORM-OBJECTS00] on [PLATFORM0] by placing them in a spiral, starting from the center of the [PLATFORM0] and expanding outward."
- "Arrange [SUB-PLATFORM-CATEGORY-OBJECTS00] on [PLATFORM0] by height and type."
- "Design a thematic organization system on [PLATFORM0] for all [SUB-PLATFORM-OBJECTS00], separating items by color and function."
- "Organize the entire contents of [PLATFORM0] by categorizing items into labeled sections, with frequently used items at a central location for easy access."
- "Declutter the right side of [PLATFORM0] by moving all loose objects to the left side and aligning them by size."
- "Design a "mirrored room" on top of [PLATFORM0] by arranging the objects to create a sense of symmetry and reflection."
- "Create an artistic arrangement on [PLATFORM0]."
- "Carefully arrange all the [SUB-PLATFORM-CATEGORY-OBJECTS00] on [PLATFORM0] in ascending order of size, leaving a clear space in the center for the [SUB-PLATFORM-OBJECTS00] from [PLATFORM0] to be placed as a decorative piece."

The full extent of MANITASKOT-200 can be found in the submitted dataset files.



Figure 9: Visualization of the 3 Scenes Used to Generate ManiTaskGen-RAS-40K.

## 577 C.2 VLM-based Voting Mechanism

578 As described in Sec. 3.3, the process of generating outcome-based tasks involves 2 steps. Firstly, we  
 579 fill the templates from MANITASKOT-200 with valid combinations of platforms and objects, and  
 580 then we employ a VLM-based voting mechanism to roll out reasonable tasks.

581 In our implementation, we use an ensemble of 3 VLMs (GPT-4o [40], Gemini-2.5-pro [41], Claude-  
 582 3.7-sonnet [42]) to assess a given task. For each VLM, we introduce the task goals and the capabilities  
 583 of the agent (navigate, pick, place), along with pictures of all platforms involved in the task. Each  
 584 VLM is required to output 1 of 3 assessment levels: feasible, partially feasible or not feasible. A task  
 585 will be included in the final dataset only if at least 2 out of the 3 VLMs vote for it as a feasible one.  
 586 The detailed prompt we use for this process is shown below:

### Prompts for Outcome-based Task Assessment

Task: [TASK\_DESCRIPTION, IMAGES\_OF\_INVOLVED\_OBJECTS\_AND\_PLATFORMS]  
 Evaluate if the given task is feasible for such a robot whose abilities involve: navigate, pick  
 up any movable object and place objects anywhere they fit on a given platform.

Assessment criteria:

1. Are all required objects present in the scene?
2. Is there sufficient free space on the target platform given the required object placement?
3. Are the required state achievable given the robot’s capabilities and the objects we have in the involved platforms?
4. Would completing the task create any unstable or physically impossible object arrangements?

Only output a single line: "Feasible", "Partially feasible", or "Not feasible". Do not output anything else.

587

## 588 D Experiment Details

### 589 D.1 Details of ManiTaskGen-RAS-40K

590 To build ManiTaskGen-RAS-40K, we choose 1 scene from ReplicaCAD, AI2THOR and SUN-  
 591 RGBD respectively. For ReplicaCAD, we select the scene *apt\_0*; For AI2THOR, we select the scene  
 592 *floorplan\_train2\_1 of RoboTHOR*; For SUN-RGBD, we choose the scene *0000885-000176498993 of*  
 593 *hotel\_uc*. Fig. 9 shows the visualization of these scenes.

594 Next, we demonstrate the detail of generating tasks using the selected scenes. In Sec. D.1.1, we  
 595 illustrate the data preprocessing pipeline; In Sec. D.1.2, we describe how we use the processed data  
 596 to generate ManiTaskGen-RAS-40K of controlled task complexity and length.

#### 597 D.1.1 Scene Data Preprocessing

598 For each given scene, our preprocessing pipeline consists of two parts: **Item Renaming** and **Removal**  
 599 **of Overlapping Objects**.



Figure 10: **Ambiguous Item Naming in ReplicaCAD.**

600 **Item Renaming.** We observe that the original object naming in certain input scenes (e.g., ReplicaCAD  
 601 and AI2THOR) is somewhat casual. As shown in Fig. 10, many books and kitchen utensils share  
 602 identical base names and are only distinguished by numerical suffixes. This naming convention  
 603 makes it difficult for the generated process-based tasks to differentiate between Level-1 and Level-2  
 604 tasks (as defined in Sec. 5.1.2). To address this, we employ a VLM to rename objects in the given  
 605 scene. The new naming scheme follows the format "{category\_name}\_{specific\_name}". When a  
 606 process-based task involves an object that shares the same {category\_name} with other objects on  
 607 the same surface, the task is classified as Level-2; otherwise, it is treated as Level-1. The prompt we  
 608 provide to the VLM is detailed below. The renaming results for ReplicaCAD are presented in Tab. 5,

#### Prompts for Item Renaming

You are a professional item classification and naming assistant. You need to provide a unique and meaningful name for each item based on indoor object images.

I need your help to classify and rename a series of indoor object images. Each image contains an item, probably on a supporting platform, to be named. Please follow these requirements:

1. Naming Rules:

- Each item name must be unique
- Names should reflect the item's category and characteristics
- Use English naming format '{category\_name}\_{specific\_name}', no non-alphabetic characters allowed

2. Naming Examples:

- A red mug -> kitchenware\_red\_mug
- A textbook -> book\_textbook
- A desk lamp -> electronics\_desk\_lamp

Please review the images and provide standardized names according to the above rules. The following images are the items to be classified, one item per image.

[Item Image]

The provided image contains an item to be renamed, probably being placed on some platform. Your response should be a single line of the new name, with '\_' connecting different words, e.g. 'utensil\_white\_soap\_dispenser' (without quotes). Don't include redundant characters, don't include characters other than alphabets and underlines, don't coincide with any previous name you've given in the [ITEM NAME LIST].

609

Name (before)	Name (after)
frl_apartment_kitchen_utensil_01	kitchenware_ceramic_lidded_container
frl_apartment_kitchen_utensil_02	bathroom_light_green_pump_bottle
frl_apartment_kitchen_utensil_03	kitchenware_spice_container
frl_apartment_kitchen_utensil_04	kitchenware_white_paper_towel_holder
frl_apartment_kitchen_utensil_05	kitchenware_navy_blue_lidded_dish
frl_apartment_kitchen_utensil_06	kitchenware_short_glass
frl_apartment_kitchen_utensil_08	kitchenware_grey_mug
frl_apartment_kitchen_utensil_09	bathroom_dark_blue_bottle
frl_apartment_knifeblock	kitchenware_wooden_knife_block
frl_apartment_lamp_01	lighting_modern_table_lamp
frl_apartment_lamp_02	lighting_blue_base_table_lamp
frl_apartment_pan_01	kitchenware_dark_blue_saucepan
frl_apartment_picture_02	decor_autumn_tree_canvas
frl_apartment_picture_03	decor_silver_photo_frame
frl_apartment_picture_04	decor_abstract_framed_print
frl_apartment_plate_01	kitchenware_white_octagonal_plate
frl_apartment_plate_02	kitchenware_wooden_cake_stand
frl_apartment_remote-control_01	electronics_black_remote_control
frl_apartment_shoebox_01	storage_dark_grey_round_box
frl_apartment_shoe_01	shoes_black_oxford_shoes
frl_apartment_shoe_02	shoes_brown_leather_oxford
frl_apartment_shoe_03	shoes_red_slingback_sandal
frl_apartment_shoe_04	shoes_blue_sneaker
frl_apartment_small_appliance_01	kitchenware_beige_coffee_grinder
frl_apartment_small_appliance_02	bathroom_grey_soap_dispenser
frl_apartment_sponge_dish	kitchenware_small_white_bowl_with_butter
frl_apartment_basket	kitchenware_beige_utensil_holder
frl_apartment_book_01	books_dark_red_hardcover
frl_apartment_book_02	books_navy_blue_hardcover
frl_apartment_book_03	books_brown_hardcover_journal
frl_apartment_book_04	books_black_hardcover
frl_apartment_book_05	electronics_white_router_box
frl_apartment_book_06	electronics_navy_blue_printer
frl_apartment_bowl_06	kitchenware_small_white_bowl
frl_apartment_bowl_07	kitchenware_white_porcelain_bowl
frl_apartment_box	storage_brown_wooden_box
frl_apartment_camera_02	electronics_white_security_camera
frl_apartment_choppingboard_02	decor_wooden_cube
frl_apartment_clock	decor_mantel_clock
frl_apartment_cup_01	kitchenware_small_white_mug
frl_apartment_cup_02	kitchenware_blue_floral_mug
frl_apartment_cup_03	kitchenware_beige_tea_cup
frl_apartment_cushion_03	furniture_light_blue_throw_pillows

Table 5: **Renaming Results of ReplicaCAD.**

**610 Removal of Overlapped Objects.** We further observe that in certain scenes (e.g., SUN-RGBD),  
**611** objects located on the same surface exhibit significant bounding box overlap. During preprocessing,  
**612** we remove such objects to ensure data cleanliness.

### **613 D.1.2 Generating Tasks of Controlled Complexity and Length**

**614** After preprocessing the input scene data, we build the scene graph and use it to generate the tasks, as  
**615** described in Sec. 3.3. We first generate atomic actions using the patterns provided in Tab. 6. After that,  
**616** we generate all possible single-step pick-and-place tasks (i.e., Level 1 and Level 2 tasks). For more  
**617** complex, longer-horizon multi-step pick-and-place tasks, we limited them to two-step sequences  
**618** and sampled 5,000 instances for ReplicaCAD and AI2THOR, while only sampling SUN-RGBD  
**619** to 500 instances (due to fewer objects and platforms in the scene of SUN-RGBD). Similarly, for

outcome-based tasks, we generated constrained quantities for different scenes, with specific numbers detailed in Tab.7.

Atomic Action Pattern	Generation Logic
Move to a named surface	If the object can fit on the surface.
Move to a location around a specific object	If there exists a place to fit this object, which only involves receptacle regions of one specific object.
Move to a location with specific direction relative to a specific object	If there exists a place to fit this object, which only involves receptacle regions of specific direction relative to the specific object.
Move to a location between two objects	If there exists a place to fit this object, and the covered receptacle regions correspond to complementary directional pairs from two different objects (e.g., left side of object A and right side of object B; front-left of object A and back-right of object B).

Table 6: Patterns and Logics for Generating Atomic Actions.

Note that, for AI2THOR scenes, no Level-2 tasks were generated since there were no instances of multiple objects belonging to the same category existing on the same surface after the renaming process. Tab. 7 shows the exact numbers of tasks we generate for each scene.

Dataset	Level 1	Level 2	Level 3	Level 4	Total
ReplicaCAD	2195	24153	5000	500	31848
AI2THOR	2218	0	5000	100	7318
SUN-RGBD	35	120	500	50	705
Total	4448	24273	10500	650	39871

Table 7: Number of Tasks at Each Level across Datasets.



## 625 D.2 Examples of Complete Benchmark Episodes

626 Below we display two full episode histories of benchmarking experiments elaborated in Sec. 5.1.3.  
627 In Sec. D.2.1, we present a successful episode performed by Claude-3.7-sonnet in the ReplicaCAD  
628 scene. In Sec. D.2.2, we show a failed episode performed by Gemini-2.5-pro in the AI2THOR scene.

### 629 D.2.1 Successful Episode

630 The following episode shows how claude-3.7-sonnet performed a task successfully in ReplicaCAD.  
631 When encountered perceptual ambiguity, it repeatedly picking up the possible objects, and immedi-  
632 ately use *place\_r* to place it down when recognized. Eventually it picks up the correct item at the  
633 fourth attempt. Then it went to the destination, asked for the receptacle pictures and placed it down  
634 successfully.



Figure 11: Images Given at the Step 0 of the Successful Episode.

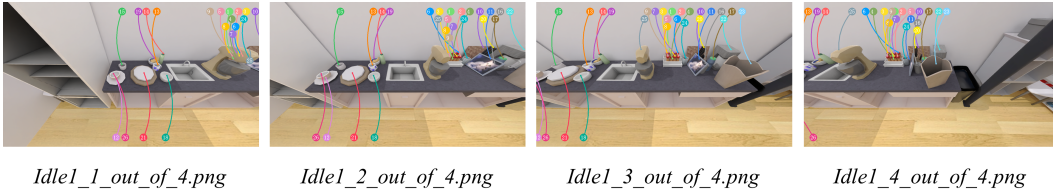


Figure 12: Images Given after Step 1 of the Successful Episode.

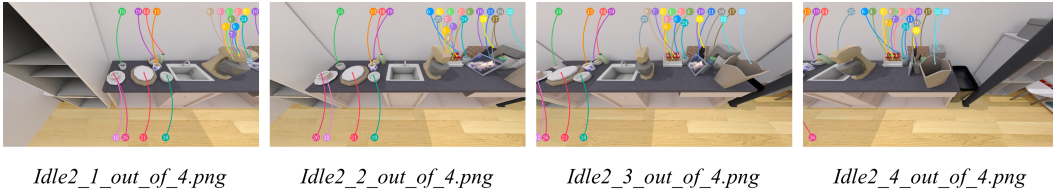


Figure 13: Images Given after Step 2 of the Successful Episode.



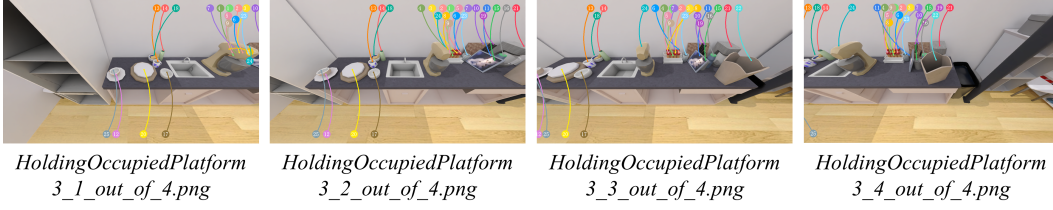


Figure 14: Images Given after Step 3 of the Successful Episode.

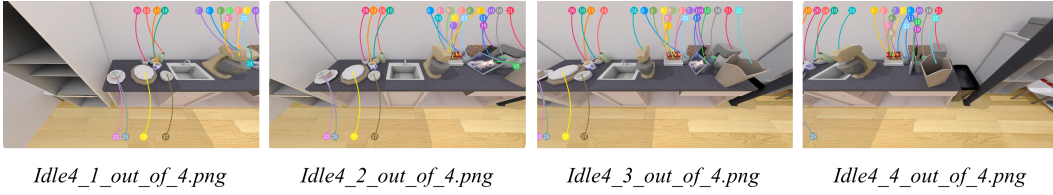


Figure 15: Images Given after Step 4 of the Successful Episode.

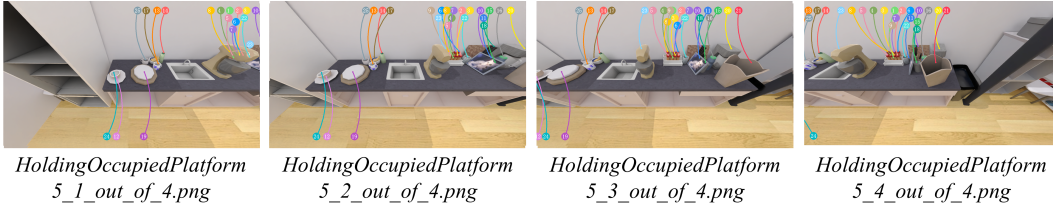


Figure 16: Images Given after Step 5 of the Successful Episode.

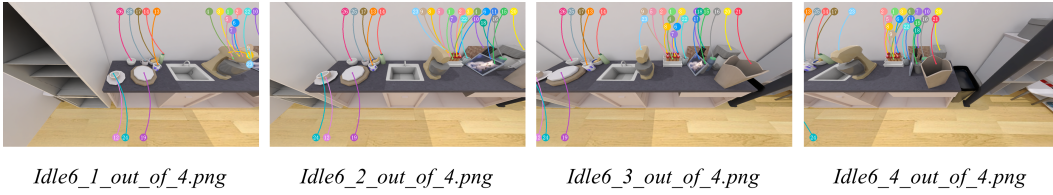


Figure 17: Images Given after Step 6 of the Successful Episode.

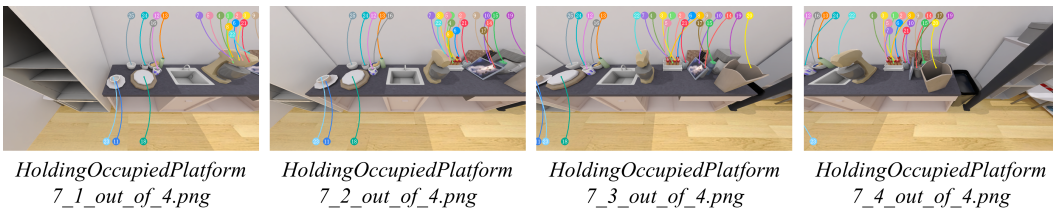


Figure 18: Images Given after Step 7 of the Successful Episode.

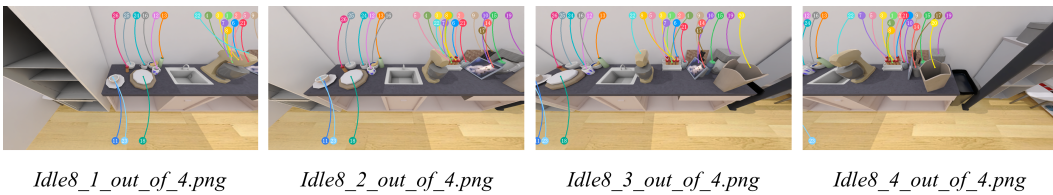


Figure 19: Images Given after Step 8 of the Successful Episode.



*HoldingOccupiedPlatform  
9\_1\_out\_of\_4.png*

*HoldingOccupiedPlatform  
9\_2\_out\_of\_4.png*

*HoldingOccupiedPlatform  
9\_3\_out\_of\_4.png*

*HoldingOccupiedPlatform  
9\_4\_out\_of\_4.png*

**Figure 20: Images Given after Step 9 of the Successful Episode.**



*HoldingOccupiedPlatform  
10\_1\_out\_of\_4.png*

*HoldingOccupiedPlatform  
10\_2\_out\_of\_4.png*

**Figure 21: Images Given after Step 10 of the Successful Episode.**



*HoldingOccupiedPlatform  
11\_1\_out\_of\_4.png*

*HoldingOccupiedPlatform  
11\_2\_out\_of\_4.png*

*ShowReceptacle11.png*

**Figure 22: Images Given after Step 11 of the Successful Episode.**



*HoldingOccupiedPlatform  
12\_out\_of\_4.png*

*HoldingOccupiedPlatform  
12\_2\_out\_of\_4.png*

*ShowReceptacle12.png*

**Figure 23: Images Given after Step 12 of the Successful Episode.**



*Idle13\_1\_out\_of\_4.png*

*Idle13\_2\_out\_of\_4.png*

**Figure 24: Images Given after Step 13 of the Successful Episode.**

### Success Episode by Claude-3.7-sonnet

System Message: You are a household assistant robot equipped with a vision system and manipulation capabilities. You need to complete a task in a 3D environment, mainly involving moving, picking up, and placing objects. Your goal is to move an object from one place to another correctly according to instructions. Please strictly follow the given Action Space for operations.

At the beginning of each task, you will be informed of a specific task to complete.

Throughout the task execution, you will transition between multiple states and have access to a set of possible actions (action space).

You may receive one of the following task types involving one object and another object or a platform. (an object/another object/a platform are the name of objects or platforms, dir is one of the 8 directions):

- move an object to a platform. - move an object to the dir of a platform. - move an object around another object. - move an object to the dir of another object. - move an object between another object and yet another object.

Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.

Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions:

[front-left, front, front-right],

[left, center, right],

[rear-left, rear, rear-right]

Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Task success criteria:

- move an object to a platform: The task is considered successful only when an object is placed anywhere on the specified platform.

- move an object to the dir of a platform: The task is considered successful only when an object is placed on the platform and the geometric center of the placement is in the dir direction relative to the platform.

- move an object around another object: The task is considered successful only when an object is placed in any of the receptacles surrounding another object.

- move an object to the dir of another object: The task is considered successful only when an object is placed in the receptacles surrounding another object and overlaps with the dir direction.

- move an object between another object and yet another object: The task is considered successful only when an object is placed such that it intersects with receptacles in a pair of opposite directions between another object and yet another object.

Your action space, includes walking to platforms, picking up items, putting items down, rotating direction, getting information about directions around an object; CALL\_END, placing an item in a specific direction on a platform, in a direction relative to another object, or placing it anywhere. Whenever you're about to choose an action, I will provide your current action space. If you choose an action that cannot be performed, such as trying to pick up an object repeatedly or attempting to place an object in a space that's too small, you'll receive a notification.

Picking up an item: Corresponds to the command "pick\_object\_{object number}\_of\_current\_platform".

Walking to a platform: Corresponds to the command "go\_to\_platform\_name".

Rotating direction: Corresponds to the command "change\_view".

Getting an image of an object: Corresponds to the command "show\_receptacle\_of\_object\_{object number}\_of\_current\_platform".

Putting down your holding item on empty platform: Corresponds to the command "place\_s\_[(region\_idx1,region\_idx2)...]" where dir\_idx1, dir\_idx2 are NUMBERS, indicating the index of the receptacles; Putting down your item on an occupied platform: Corresponds to the command "place\_s\_[(object\_idx1,dir\_idx1)...]", where object\_idx1, dir\_idx1 are NUMBERS, indicating the object and its corresponding receptacles. For example, "place\_s\_[(1,2), (2,5)]" means putting down your item at the union of object 1's 2nd receptacles and object 2's 5th receptacles. Note that for placement action, the system will attempt to find a valid placement area that intersects with all the regions you selected. If it fails, it will place the object anywhere within your chosen space and you'll receive a notification.

You may be shown various types of images:

0) At the beginning of a task, if the task involves objects on a platform that contains other objects of the same type (e.g., the object is a book on a bookshelf with many books), you will receive a notification and images of these objects to help you distinguish them.

1) Whenever you move to a platform, you will see an image of the platform from your current perspective (or multiple images for longer platforms, indicating you can move horizontally to better observe the platform). If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects. If there are objects on the platform, guiding lines will indicate the object numbers.

2) If there are objects on a platform, you can use "show\_receptable\_of\_object\_x\_of\_current\_platform" to get a focused image of that object with numbered receptacles around it. This allows you to see the available receptacles around the object.

Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first (although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

[\*See Fig. 11]

**The task involves identical objects. To help you distinguish them, we have provided images of the relevant objects.**

Here are 1 images, named [AmbiguousObject.png'] respectively. Their description is as follows: This image, AmbiguousObject.png, is an image of the object you are going to move, with its adjacent objects transparentized. The object is kitchenware\_white\_paper\_towel\_holder. The object is located on object\_kitchen\_counter\_1\_body\_platform\_1.

Here are 1 images, named [AmbiguousDestinationObjectA.png'] respectively. Their description is as follows: This image, AmbiguousDestinationObjectA.png, is an image of the object you are going to place an object nearby, with its adjacent objects transparentized. The object is lighting\_blue\_base\_table\_lamp. The object is located on object\_table\_02\_40\_platform\_2.

Here are 1 images, named [AmbiguousDestinationObjectB.png'] respectively. Their description is as follows: This image, AmbiguousDestinationObjectB.png, is an image of the object you are going to place an object nearby, with its adjacent objects transparentized. The object is lighting\_blue\_base\_table\_lamp. The object is located on object\_table\_02\_40\_platform\_2.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Initially, kitchenware\_white\_paper\_towel\_holder is on

object\_kitchen\_counter\_1\_body\_platform\_1, and lighting\_blue\_base\_table\_lamp is on object\_table\_02\_40\_platform\_2, together with lighting\_blue\_base\_table\_lamp.

Steps used: 0/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

Your available action space:

- 'CALL\_END': Call end (terminate the task). - ['go\_to\_object\_sofa\_10\_platform\_0', 'go\_to\_object\_sofa\_10\_platform\_1', 'go\_to\_object\_sofa\_10\_platform\_2', 'go\_to\_object\_table\_04\_13\_platform\_0', 'go\_to\_object\_chair\_04\_46\_platform\_0', 'go\_to\_object\_chair\_04\_47\_platform\_0', 'go\_to\_object\_chair\_05\_7\_platform\_0', 'go\_to\_object\_chair\_05\_8\_platform\_0', 'go\_to\_object\_stool\_02\_18\_platform\_0', 'go\_to\_object\_stool\_02\_6\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_1', 'go\_to\_object\_rack\_01\_75\_platform\_2', 'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_0', 'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_3', 'go\_to\_object\_fridge\_0\_body\_platform\_4', 'go\_to\_object\_fridge\_0\_body\_platform\_5', 'go\_to\_object\_fridge\_0\_body\_platform\_6', 'go\_to\_object\_fridge\_0\_body\_platform\_7', 'go\_to\_object\_bin\_03\_3\_platform\_0', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_0', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_1', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_2', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_3', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_4', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_5', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_6', 'go\_to\_object\_table\_03\_14\_platform\_1', 'go\_to\_object\_table\_01\_48\_platform\_0', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_0', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_1', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_2', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_3', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_4', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_5', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_6', 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_0', 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_1', 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_2', 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_3', 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_4', 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_5', 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_6', 'go\_to\_object\_table\_02\_40\_platform\_0', 'go\_to\_object\_table\_02\_40\_platform\_1', 'go\_to\_object\_table\_02\_40\_platform\_2', 'go\_to\_object\_tvstand\_88\_platform\_0', 'go\_to\_object\_cabinet\_3\_body\_platform\_0', 'go\_to\_object\_cabinet\_3\_body\_platform\_1'].

Please choose one of the available actions.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

-----  
VLM Response: go\_to\_object\_kitchen\_counter\_1\_body\_platform\_1

-----  
User Message:

You are currently at object\_kitchen\_counter\_1\_body\_platform\_1.

And you are holding nothing.

Your available action space:

- ['go\_to\_object\_sofa\_10\_platform\_0', 'go\_to\_object\_sofa\_10\_platform\_1', 'go\_to\_object\_sofa\_10\_platform\_2', 'go\_to\_object\_table\_04\_13\_platform\_0', 'go\_to\_object\_chair\_04\_46\_platform\_0', 'go\_to\_object\_chair\_04\_47\_platform\_0',



```

'go_to_object_chair_05_7_platform_0', 'go_to_object_chair_05_8_platform_0',
'go_to_object_stool_02_18_platform_0', 'go_to_object_stool_02_6_platform_0',
'go_to_object_rack_01_75_platform_0', 'go_to_object_rack_01_75_platform_1',
'go_to_object_rack_01_75_platform_2',
'go_to_object_kitchen_counter_1_body_platform_0',
'go_to_object_kitchen_counter_1_body_platform_1',
'go_to_object_fridge_0_body_platform_1', 'go_to_object_fridge_0_body_platform_3',
'go_to_object_fridge_0_body_platform_4', 'go_to_object_fridge_0_body_platform_5',
'go_to_object_fridge_0_body_platform_6', 'go_to_object_fridge_0_body_platform_7',
'go_to_object_bin_03_3_platform_0', 'go_to_object_wall_cabinet_01_4_platform_0',
'go_to_object_wall_cabinet_01_4_platform_1',
'go_to_object_wall_cabinet_01_4_platform_2',
'go_to_object_wall_cabinet_01_4_platform_3',
'go_to_object_wall_cabinet_01_4_platform_4',
'go_to_object_wall_cabinet_01_4_platform_5',
'go_to_object_wall_cabinet_01_4_platform_6', 'go_to_object_table_03_14_platform_1',
'go_to_object_table_01_48_platform_0', 'go_to_object_wall_cabinet_02_21_platform_0',
'go_to_object_wall_cabinet_02_21_platform_1',
'go_to_object_wall_cabinet_02_21_platform_2',
'go_to_object_wall_cabinet_02_21_platform_3',
'go_to_object_wall_cabinet_02_21_platform_4',
'go_to_object_wall_cabinet_02_21_platform_5',
'go_to_object_wall_cabinet_02_21_platform_6',
'go_to_object_chestOfDrawers_01_2_body_platform_0',
'go_to_object_chestOfDrawers_01_2_body_platform_1',
'go_to_object_chestOfDrawers_01_2_body_platform_2',
'go_to_object_chestOfDrawers_01_2_body_platform_3',
'go_to_object_chestOfDrawers_01_2_body_platform_4',
'go_to_object_chestOfDrawers_01_2_body_platform_5',
'go_to_object_chestOfDrawers_01_2_body_platform_6',
'go_to_object_table_02_40_platform_0', 'go_to_object_table_02_40_platform_1',
'go_to_object_table_02_40_platform_2', 'go_to_object_tvstand_88_platform_0',
'go_to_object_cabinet_3_body_platform_0', 'go_to_object_cabinet_3_body_platform_1'] -
['pick_object_1_of_current_platform', 'pick_object_2_of_current_platform',
'pick_object_3_of_current_platform', 'pick_object_4_of_current_platform',
'pick_object_5_of_current_platform', 'pick_object_6_of_current_platform',
'pick_object_7_of_current_platform', 'pick_object_8_of_current_platform',
'pick_object_9_of_current_platform', 'pick_object_10_of_current_platform',
'pick_object_11_of_current_platform', 'pick_object_12_of_current_platform',
'pick_object_13_of_current_platform', 'pick_object_14_of_current_platform',
'pick_object_15_of_current_platform', 'pick_object_16_of_current_platform',
'pick_object_17_of_current_platform', 'pick_object_18_of_current_platform',
'pick_object_19_of_current_platform', 'pick_object_20_of_current_platform',
'pick_object_21_of_current_platform', 'pick_object_22_of_current_platform',
'pick_object_23_of_current_platform', 'pick_object_24_of_current_platform',
'pick_object_25_of_current_platform', 'pick_object_26_of_current_platform'] -
['show_receptacle_of_object_1_of_current_platform',
'show_receptacle_of_object_2_of_current_platform',
'show_receptacle_of_object_3_of_current_platform',
'show_receptacle_of_object_4_of_current_platform',
'show_receptacle_of_object_5_of_current_platform',
'show_receptacle_of_object_6_of_current_platform',
'show_receptacle_of_object_7_of_current_platform',
'show_receptacle_of_object_8_of_current_platform',
'show_receptacle_of_object_9_of_current_platform',
'show_receptacle_of_object_10_of_current_platform',
'show_receptacle_of_object_11_of_current_platform',

```

'show\_receptacle\_of\_object\_12\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_13\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_14\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_15\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_16\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_17\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_18\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_19\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_20\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_21\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_22\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_23\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_24\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_25\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_26\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'change\_view': Change the viewing angle to observe the scene from another perspective.

Choose one of the available actions.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 1/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 12]

Here are 4 images, named ['Idle\_1\_1\_out\_of\_4.png', 'Idle\_1\_2\_out\_of\_4.png', 'Idle\_1\_3\_out\_of\_4.png', 'Idle\_1\_4\_out\_of\_4.png'] respectively. Their description is as follows: These images show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_kitchen\_counter\_1\_body\_platform\_1. If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
 VLM Response: change\_view

-----  
 User Message:

**Unable to rotate to another view. The platform you at only have 1 walkable place for you.**

You are currently at object\_kitchen\_counter\_1\_body\_platform\_1.

And you are holding nothing.

Your available action space:

```
- ['go_to_object_sofa_10_platform_0', 'go_to_object_sofa_10_platform_1',  
'go_to_object_sofa_10_platform_2', 'go_to_object_table_04_13_platform_0',  
'go_to_object_chair_04_46_platform_0', 'go_to_object_chair_04_47_platform_0',  
'go_to_object_chair_05_7_platform_0', 'go_to_object_chair_05_8_platform_0',  
'go_to_object_stool_02_18_platform_0', 'go_to_object_stool_02_6_platform_0',  
'go_to_object_rack_01_75_platform_0', 'go_to_object_rack_01_75_platform_1',  
'go_to_object_rack_01_75_platform_2',  
'go_to_object_kitchen_counter_1_body_platform_0',  
'go_to_object_kitchen_counter_1_body_platform_1',  
'go_to_object_fridge_0_body_platform_1', 'go_to_object_fridge_0_body_platform_3',  
'go_to_object_fridge_0_body_platform_4', 'go_to_object_fridge_0_body_platform_5',  
'go_to_object_fridge_0_body_platform_6', 'go_to_object_fridge_0_body_platform_7',  
'go_to_object_bin_03_3_platform_0', 'go_to_object_wall_cabinet_01_4_platform_0',  
'go_to_object_wall_cabinet_01_4_platform_1',  
'go_to_object_wall_cabinet_01_4_platform_2',  
'go_to_object_wall_cabinet_01_4_platform_3',  
'go_to_object_wall_cabinet_01_4_platform_4',  
'go_to_object_wall_cabinet_01_4_platform_5',  
'go_to_object_wall_cabinet_01_4_platform_6', 'go_to_object_table_03_14_platform_1',  
'go_to_object_table_01_48_platform_0', 'go_to_object_wall_cabinet_02_21_platform_0',  
'go_to_object_wall_cabinet_02_21_platform_1',  
'go_to_object_wall_cabinet_02_21_platform_2',  
'go_to_object_wall_cabinet_02_21_platform_3',  
'go_to_object_wall_cabinet_02_21_platform_4',  
'go_to_object_wall_cabinet_02_21_platform_5',  
'go_to_object_wall_cabinet_02_21_platform_6',  
'go_to_object_chestOfDrawers_01_2_body_platform_0',  
'go_to_object_chestOfDrawers_01_2_body_platform_1',  
'go_to_object_chestOfDrawers_01_2_body_platform_2',  
'go_to_object_chestOfDrawers_01_2_body_platform_3',  
'go_to_object_chestOfDrawers_01_2_body_platform_4',  
'go_to_object_chestOfDrawers_01_2_body_platform_5',  
'go_to_object_chestOfDrawers_01_2_body_platform_6',  
'go_to_object_table_02_40_platform_0', 'go_to_object_table_02_40_platform_1',  
'go_to_object_table_02_40_platform_2', 'go_to_object_tvstand_88_platform_0',  
'go_to_object_cabinet_3_body_platform_0', 'go_to_object_cabinet_3_body_platform_1'] -  
['pick_object_1_of_current_platform', 'pick_object_2_of_current_platform',  
'pick_object_3_of_current_platform', 'pick_object_4_of_current_platform',  
'pick_object_5_of_current_platform', 'pick_object_6_of_current_platform',  
'pick_object_7_of_current_platform', 'pick_object_8_of_current_platform',  
'pick_object_9_of_current_platform', 'pick_object_10_of_current_platform',  
'pick_object_11_of_current_platform', 'pick_object_12_of_current_platform',  
'pick_object_13_of_current_platform', 'pick_object_14_of_current_platform',  
'pick_object_15_of_current_platform', 'pick_object_16_of_current_platform',  
'pick_object_17_of_current_platform', 'pick_object_18_of_current_platform',  
'pick_object_19_of_current_platform', 'pick_object_20_of_current_platform',  
'pick_object_21_of_current_platform', 'pick_object_22_of_current_platform',  
'pick_object_23_of_current_platform', 'pick_object_24_of_current_platform',  
'pick_object_25_of_current_platform', 'pick_object_26_of_current_platform'] -  
['show_receptacle_of_object_1_of_current_platform',  
'show_receptacle_of_object_2_of_current_platform',  
'show_receptacle_of_object_3_of_current_platform',  
'show_receptacle_of_object_4_of_current_platform',  
'show_receptacle_of_object_5_of_current_platform',
```



'show\_receptacle\_of\_object\_6\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_7\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_8\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_9\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_10\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_11\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_12\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_13\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_14\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_15\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_16\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_17\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_18\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_19\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_20\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_21\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_22\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_23\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_24\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_25\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_26\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'change\_view': Change the viewing angle to observe the scene from another perspective.

Choose one of the available actions.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 2/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 13]

Here are 4 images, named ['Idle\_2\_1\_out\_of\_4.png', 'Idle\_2\_2\_out\_of\_4.png', 'Idle\_2\_3\_out\_of\_4.png', 'Idle\_2\_4\_out\_of\_4.png'] respectively.

Their description is as follows: These images show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_kitchen\_counter\_1\_body\_platform\_1.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you

think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM\_response: pick\_15\_of\_current\_platform  
-----

User Message:

You are currently at object\_kitchen\_counter\_1\_body\_platform\_1.

And you are holding kitchenware\_small\_white\_mug.

Your available action space:

-['go\_to\_object\_sofa\_10\_platform\_0', 'go\_to\_object\_sofa\_10\_platform\_1',  
'go\_to\_object\_sofa\_10\_platform\_2', 'go\_to\_object\_table\_04\_13\_platform\_0',  
'go\_to\_object\_chair\_04\_46\_platform\_0', 'go\_to\_object\_chair\_04\_47\_platform\_0',  
'go\_to\_object\_chair\_05\_7\_platform\_0', 'go\_to\_object\_chair\_05\_8\_platform\_0',  
'go\_to\_object\_stool\_02\_18\_platform\_0', 'go\_to\_object\_stool\_02\_6\_platform\_0',  
'go\_to\_object\_rack\_01\_75\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_1',  
'go\_to\_object\_rack\_01\_75\_platform\_2',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_0',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_1',  
'go\_to\_object\_fridge\_0\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_3',  
'go\_to\_object\_fridge\_0\_body\_platform\_4', 'go\_to\_object\_fridge\_0\_body\_platform\_5',  
'go\_to\_object\_fridge\_0\_body\_platform\_6', 'go\_to\_object\_fridge\_0\_body\_platform\_7',  
'go\_to\_object\_bin\_03\_3\_platform\_0', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_6', 'go\_to\_object\_table\_03\_14\_platform\_1',  
'go\_to\_object\_table\_01\_48\_platform\_0', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_6',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_0',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_1',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_2',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_3',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_4',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_5',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_6',  
'go\_to\_object\_table\_02\_40\_platform\_0', 'go\_to\_object\_table\_02\_40\_platform\_1',  
'go\_to\_object\_table\_02\_40\_platform\_2', 'go\_to\_object\_tvstand\_88\_platform\_0',  
'go\_to\_object\_cabinet\_3\_body\_platform\_0', 'go\_to\_object\_cabinet\_3\_body\_platform\_1'] -  
['show\_receptacle\_of\_object\_1\_of\_current\_platform',  
'show\_receptacle\_of\_object\_2\_of\_current\_platform',  
'show\_receptacle\_of\_object\_3\_of\_current\_platform',  
'show\_receptacle\_of\_object\_4\_of\_current\_platform',  
'show\_receptacle\_of\_object\_5\_of\_current\_platform',  
'show\_receptacle\_of\_object\_6\_of\_current\_platform',  
'show\_receptacle\_of\_object\_7\_of\_current\_platform',  
'show\_receptacle\_of\_object\_8\_of\_current\_platform',  
'show\_receptacle\_of\_object\_9\_of\_current\_platform',  
'show\_receptacle\_of\_object\_10\_of\_current\_platform',  
'show\_receptacle\_of\_object\_11\_of\_current\_platform',  
'show\_receptacle\_of\_object\_12\_of\_current\_platform',

'show\_receptacle\_of\_object\_13\_of\_current\_platform',  
'show\_receptacle\_of\_object\_14\_of\_current\_platform',  
'show\_receptacle\_of\_object\_15\_of\_current\_platform',  
'show\_receptacle\_of\_object\_16\_of\_current\_platform',  
'show\_receptacle\_of\_object\_17\_of\_current\_platform',  
'show\_receptacle\_of\_object\_18\_of\_current\_platform',  
'show\_receptacle\_of\_object\_19\_of\_current\_platform',  
'show\_receptacle\_of\_object\_20\_of\_current\_platform',  
'show\_receptacle\_of\_object\_21\_of\_current\_platform',  
'show\_receptacle\_of\_object\_22\_of\_current\_platform',  
'show\_receptacle\_of\_object\_23\_of\_current\_platform',  
'show\_receptacle\_of\_object\_24\_of\_current\_platform',  
'show\_receptacle\_of\_object\_25\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'change\_view': Change the viewing angle to observe the scene from another perspective. - 'place\_r': Place the object in an arbitrary place on the platform. - 'place\_s [(object\_idx1,dir\_idx1)...]': Place the object in the union area of regions with indices (object\_idx1,region\_idx1)(object\_idx1's region\_idx1 space) etc. All available pairs are in [(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (5, 1), (6, 1), (8, 1), (10, 1), (11, 1), (11, 2), (11, 3), (12, 1), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (12, 7), (12, 8), (13, 1), (14, 1), (14, 2), (14, 3), (14, 4), (14, 5), (16, 1), (16, 2), (16, 3), (17, 1), (17, 2), (17, 3), (17, 4), (17, 5), (17, 6), (18, 1), (19, 1), (20, 1), (20, 2), (20, 3), (20, 4), (20, 5), (21, 1), (21, 2), (21, 3), (21, 4), (21, 5), (22, 1), (22, 2), (22, 3), (23, 1), (24, 1), (24, 2), (24, 3), (24, 4), (24, 5), (24, 6), (24, 7), (24, 8), (25, 1), (25, 2)], and a valid choice should be a subset of it.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.

Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions: [front-left, front, front-right], [left, center, right], [rear-left, rear, rear-right]

Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:  
Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp  
Steps used: 3/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 14]

Here are 4 images, named ['HoldingOccupiedPlatform3\_1\_out\_of\_4.png',  
'HoldingOccupiedPlatform3\_2\_out\_of\_4.png',  
'HoldingOccupiedPlatform3\_3\_out\_of\_4.png',  
'HoldingOccupiedPlatform3\_4\_out\_of\_4.png'] respectively.

Their description is as follows: These image show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_kitchen\_counter\_1\_body\_platform\_1.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will be marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM Response: place\_r

-----  
User Message:

**The object has been placed successfully. You can now call end if you think you've finished the task correctly, or can also pick up the object again if you think the placement is not correct.**

You are currently at object\_kitchen\_counter\_1\_body\_platform\_1.

And you are holding nothing.

Your available action space:

- ['go\_to\_object\_sofa\_10\_platform\_0', 'go\_to\_object\_sofa\_10\_platform\_1',  
'go\_to\_object\_sofa\_10\_platform\_2', 'go\_to\_object\_table\_04\_13\_platform\_0',  
'go\_to\_object\_chair\_04\_46\_platform\_0', 'go\_to\_object\_chair\_04\_47\_platform\_0',  
'go\_to\_object\_chair\_05\_7\_platform\_0', 'go\_to\_object\_chair\_05\_8\_platform\_0',  
'go\_to\_object\_stool\_02\_18\_platform\_0', 'go\_to\_object\_stool\_02\_6\_platform\_0',  
'go\_to\_object\_rack\_01\_75\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_1',  
'go\_to\_object\_rack\_01\_75\_platform\_2',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_0',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_1',  
'go\_to\_object\_fridge\_0\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_3',  
'go\_to\_object\_fridge\_0\_body\_platform\_4', 'go\_to\_object\_fridge\_0\_body\_platform\_5',  
'go\_to\_object\_fridge\_0\_body\_platform\_6', 'go\_to\_object\_fridge\_0\_body\_platform\_7',  
'go\_to\_object\_bin\_03\_3\_platform\_0', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_6', 'go\_to\_object\_table\_03\_14\_platform\_1',  
'go\_to\_object\_table\_01\_48\_platform\_0', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_6',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_0',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_1',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_2',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_3',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_4',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_5',

```
'go_to_object_chestOfDrawers_01_2_body_platform_6',
'go_to_object_table_02_40_platform_0', 'go_to_object_table_02_40_platform_1',
'go_to_object_table_02_40_platform_2', 'go_to_object_tvstand_88_platform_0',
'go_to_object_cabinet_3_body_platform_0', 'go_to_object_cabinet_3_body_platform_1'] -
['pick_object_1_of_current_platform', 'pick_object_2_of_current_platform',
'pick_object_3_of_current_platform', 'pick_object_4_of_current_platform',
'pick_object_5_of_current_platform', 'pick_object_6_of_current_platform',
'pick_object_7_of_current_platform', 'pick_object_8_of_current_platform',
'pick_object_9_of_current_platform', 'pick_object_10_of_current_platform',
'pick_object_11_of_current_platform', 'pick_object_12_of_current_platform',
'pick_object_13_of_current_platform', 'pick_object_14_of_current_platform',
'pick_object_15_of_current_platform', 'pick_object_16_of_current_platform',
'pick_object_17_of_current_platform', 'pick_object_18_of_current_platform',
'pick_object_19_of_current_platform', 'pick_object_20_of_current_platform',
'pick_object_21_of_current_platform', 'pick_object_22_of_current_platform',
'pick_object_23_of_current_platform', 'pick_object_24_of_current_platform',
'pick_object_25_of_current_platform', 'pick_object_26_of_current_platform'] -
['show_receptacle_of_object_1_of_current_platform',
'show_receptacle_of_object_2_of_current_platform',
'show_receptacle_of_object_3_of_current_platform',
'show_receptacle_of_object_4_of_current_platform',
'show_receptacle_of_object_5_of_current_platform',
'show_receptacle_of_object_6_of_current_platform',
'show_receptacle_of_object_7_of_current_platform',
'show_receptacle_of_object_8_of_current_platform',
'show_receptacle_of_object_9_of_current_platform',
'show_receptacle_of_object_10_of_current_platform',
'show_receptacle_of_object_11_of_current_platform',
'show_receptacle_of_object_12_of_current_platform',
'show_receptacle_of_object_13_of_current_platform',
'show_receptacle_of_object_14_of_current_platform',
'show_receptacle_of_object_15_of_current_platform',
'show_receptacle_of_object_16_of_current_platform',
'show_receptacle_of_object_17_of_current_platform',
'show_receptacle_of_object_18_of_current_platform',
'show_receptacle_of_object_19_of_current_platform',
'show_receptacle_of_object_20_of_current_platform',
'show_receptacle_of_object_21_of_current_platform',
'show_receptacle_of_object_22_of_current_platform',
'show_receptacle_of_object_23_of_current_platform',
'show_receptacle_of_object_24_of_current_platform',
'show_receptacle_of_object_25_of_current_platform',
'show_receptacle_of_object_26_of_current_platform'] - 'CALL_END': Call end (terminate
the task). - 'change_view': Change the viewing angle to observe the scene from another
perspective.
```

Choose one of the available actions.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.

3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.

4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 4/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 15]

Here are 4 images, named ['Idle\_4\_1\_out\_of\_4.png', 'Idle\_4\_2\_out\_of\_4.png', 'Idle\_4\_3\_out\_of\_4.png', 'Idle\_4\_4\_out\_of\_4.png'] respectively.

Their description is as follows: These images show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_kitchen\_counter\_1\_body\_platform\_1.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will be marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM\_response: pick\_object\_17\_of\_current\_platform  
-----

User Message:

You are currently at object\_kitchen\_counter\_1\_body\_platform\_1.

And you are holding kitchenware\_small\_white\_bowl.

Your available actions: -['go\_to\_object\_sofa\_10\_platform\_0',  
'go\_to\_object\_sofa\_10\_platform\_1', 'go\_to\_object\_sofa\_10\_platform\_2',  
'go\_to\_object\_table\_04\_13\_platform\_0', 'go\_to\_object\_chair\_04\_46\_platform\_0',  
'go\_to\_object\_chair\_04\_47\_platform\_0', 'go\_to\_object\_chair\_05\_7\_platform\_0',  
'go\_to\_object\_chair\_05\_8\_platform\_0', 'go\_to\_object\_stool\_02\_18\_platform\_0',  
'go\_to\_object\_stool\_02\_6\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_0',  
'go\_to\_object\_rack\_01\_75\_platform\_1', 'go\_to\_object\_rack\_01\_75\_platform\_2',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_0',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_1',  
'go\_to\_object\_fridge\_0\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_3',  
'go\_to\_object\_fridge\_0\_body\_platform\_4', 'go\_to\_object\_fridge\_0\_body\_platform\_5',  
'go\_to\_object\_fridge\_0\_body\_platform\_6', 'go\_to\_object\_fridge\_0\_body\_platform\_7',  
'go\_to\_object\_bin\_03\_3\_platform\_0', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_6', 'go\_to\_object\_table\_03\_14\_platform\_1',  
'go\_to\_object\_table\_01\_48\_platform\_0', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_6',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_0',

```

'go_to_object_chestOfDrawers_01_2_body_platform_1',
'go_to_object_chestOfDrawers_01_2_body_platform_2',
'go_to_object_chestOfDrawers_01_2_body_platform_3',
'go_to_object_chestOfDrawers_01_2_body_platform_4',
'go_to_object_chestOfDrawers_01_2_body_platform_5',
'go_to_object_chestOfDrawers_01_2_body_platform_6',
'go_to_object_table_02_40_platform_0', 'go_to_object_table_02_40_platform_1',
'go_to_object_table_02_40_platform_2', 'go_to_object_tvstand_88_platform_0',
'go_to_object_cabinet_3_body_platform_0', 'go_to_object_cabinet_3_body_platform_1'] -
['show_receptacle_of_object_1_of_current_platform',
'show_receptacle_of_object_2_of_current_platform',
'show_receptacle_of_object_3_of_current_platform',
'show_receptacle_of_object_4_of_current_platform',
'show_receptacle_of_object_5_of_current_platform',
'show_receptacle_of_object_6_of_current_platform',
'show_receptacle_of_object_7_of_current_platform',
'show_receptacle_of_object_8_of_current_platform',
'show_receptacle_of_object_9_of_current_platform',
'show_receptacle_of_object_10_of_current_platform',
'show_receptacle_of_object_11_of_current_platform',
'show_receptacle_of_object_12_of_current_platform',
'show_receptacle_of_object_13_of_current_platform',
'show_receptacle_of_object_14_of_current_platform',
'show_receptacle_of_object_15_of_current_platform',
'show_receptacle_of_object_16_of_current_platform',
'show_receptacle_of_object_17_of_current_platform',
'show_receptacle_of_object_18_of_current_platform',
'show_receptacle_of_object_19_of_current_platform',
'show_receptacle_of_object_20_of_current_platform',
'show_receptacle_of_object_21_of_current_platform',
'show_receptacle_of_object_22_of_current_platform',
'show_receptacle_of_object_23_of_current_platform',
'show_receptacle_of_object_24_of_current_platform',
'show_receptacle_of_object_25_of_current_platform'] - 'CALL_END': Call end (terminate
the task). - 'change_view': Change the viewing angle to observe the scene from another
perspective. - 'place_r': Place the object in an arbitrary place on the platform. -
'place_s_[(object_idx1,dir_idx1)...]': Place the object in the union area of regions with
indices (object_idx1,region_idx1)(object_idx1's region_idx1 space) etc. All available pairs
are in [(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (5, 1), (6, 1), (8, 1), (10, 1), (11, 1), (11, 2), (11, 3),
(12, 1), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (12, 7), (12, 8), (13, 1), (13, 2), (14, 1), (14, 2),
(14, 3), (14, 4), (14, 5), (16, 1), (16, 2), (16, 3), (17, 1), (18, 1), (19, 1), (19, 2), (19, 3), (19, 4),
(19, 5), (20, 1), (20, 2), (20, 3), (20, 4), (20, 5), (21, 1), (21, 2), (21, 3), (22, 1), (23, 1), (23, 2),
(23, 3), (23, 4), (23, 5), (23, 6), (23, 7), (23, 8), (24, 1), (24, 2), (25, 1), (25, 2), (25, 3)], and a
valid choice should be a subset of it.
All your current available actions are listed as above (all are without quotes and square
brackets). Please don't mix up it with the action space of previous steps.
Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative
to the robot's perspective and the heading of the object or platform you are interacting with.
The heading of a ground object is not fixed and depends on where the robot is standing and
facing.
Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided
into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to
you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions
proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right,"
and "front-right." The grid is conceptually numbered as follows, based on these directions:
[front-left, front, front-right], [left, center, right], [rear-left, rear, rear-right]

```

Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 5/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 16]

Here are 4 images, named ['HoldingOccupiedPlatform5\_1\_out\_of\_4.png', 'HoldingOccupiedPlatform5\_2\_out\_of\_4.png', 'HoldingOccupiedPlatform5\_3\_out\_of\_4.png', 'HoldingOccupiedPlatform5\_4\_out\_of\_4.png'] respectively.

Their description is as follows: These images show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_kitchen\_counter\_1\_body\_platform\_1.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will be marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM Response: place\_r

-----  
User Message:

**The object has been placed successfully. You can now call end if you think you've finished the task correctly, or can also pick up the object again if you think the placement is not correct.**

You are currently at object\_kitchen\_counter\_1\_body\_platform\_1.

And you are holding nothing.

Your available action space: - ['go\_to\_object\_sofa\_10\_platform\_0', 'go\_to\_object\_sofa\_10\_platform\_1', 'go\_to\_object\_sofa\_10\_platform\_2', 'go\_to\_object\_table\_04\_13\_platform\_0', 'go\_to\_object\_chair\_04\_46\_platform\_0', 'go\_to\_object\_chair\_04\_47\_platform\_0', 'go\_to\_object\_chair\_05\_7\_platform\_0', 'go\_to\_object\_chair\_05\_8\_platform\_0', 'go\_to\_object\_stool\_02\_18\_platform\_0', 'go\_to\_object\_stool\_02\_6\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_1', 'go\_to\_object\_rack\_01\_75\_platform\_2', 'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_0', 'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_3', 'go\_to\_object\_fridge\_0\_body\_platform\_4', 'go\_to\_object\_fridge\_0\_body\_platform\_5',



```

'go_to_object_fridge_0_body_platform_6', 'go_to_object_fridge_0_body_platform_7',
'go_to_object_bin_03_3_platform_0', 'go_to_object_wall_cabinet_01_4_platform_0',
'go_to_object_wall_cabinet_01_4_platform_1',
'go_to_object_wall_cabinet_01_4_platform_2',
'go_to_object_wall_cabinet_01_4_platform_3',
'go_to_object_wall_cabinet_01_4_platform_4',
'go_to_object_wall_cabinet_01_4_platform_5',
'go_to_object_wall_cabinet_01_4_platform_6', 'go_to_object_table_03_14_platform_1',
'go_to_object_table_01_48_platform_0', 'go_to_object_wall_cabinet_02_21_platform_0',
'go_to_object_wall_cabinet_02_21_platform_1',
'go_to_object_wall_cabinet_02_21_platform_2',
'go_to_object_wall_cabinet_02_21_platform_3',
'go_to_object_wall_cabinet_02_21_platform_4',
'go_to_object_wall_cabinet_02_21_platform_5',
'go_to_object_wall_cabinet_02_21_platform_6',
'go_to_object_chestOfDrawers_01_2_body_platform_0',
'go_to_object_chestOfDrawers_01_2_body_platform_1',
'go_to_object_chestOfDrawers_01_2_body_platform_2',
'go_to_object_chestOfDrawers_01_2_body_platform_3',
'go_to_object_chestOfDrawers_01_2_body_platform_4',
'go_to_object_chestOfDrawers_01_2_body_platform_5',
'go_to_object_chestOfDrawers_01_2_body_platform_6',
'go_to_object_table_02_40_platform_0', 'go_to_object_table_02_40_platform_1',
'go_to_object_table_02_40_platform_2', 'go_to_object_tvstand_88_platform_0',
'go_to_object_cabinet_3_body_platform_0', 'go_to_object_cabinet_3_body_platform_1'] -
['pick_object_1_of_current_platform', 'pick_object_2_of_current_platform',
'pick_object_3_of_current_platform', 'pick_object_4_of_current_platform',
'pick_object_5_of_current_platform', 'pick_object_6_of_current_platform',
'pick_object_7_of_current_platform', 'pick_object_8_of_current_platform',
'pick_object_9_of_current_platform', 'pick_object_10_of_current_platform',
'pick_object_11_of_current_platform', 'pick_object_12_of_current_platform',
'pick_object_13_of_current_platform', 'pick_object_14_of_current_platform',
'pick_object_15_of_current_platform', 'pick_object_16_of_current_platform',
'pick_object_17_of_current_platform', 'pick_object_18_of_current_platform',
'pick_object_19_of_current_platform', 'pick_object_20_of_current_platform',
'pick_object_21_of_current_platform', 'pick_object_22_of_current_platform',
'pick_object_23_of_current_platform', 'pick_object_24_of_current_platform',
'pick_object_25_of_current_platform', 'pick_object_26_of_current_platform'] -
['show_receptacle_of_object_1_of_current_platform',
'show_receptacle_of_object_2_of_current_platform',
'show_receptacle_of_object_3_of_current_platform',
'show_receptacle_of_object_4_of_current_platform',
'show_receptacle_of_object_5_of_current_platform',
'show_receptacle_of_object_6_of_current_platform',
'show_receptacle_of_object_7_of_current_platform',
'show_receptacle_of_object_8_of_current_platform',
'show_receptacle_of_object_9_of_current_platform',
'show_receptacle_of_object_10_of_current_platform',
'show_receptacle_of_object_11_of_current_platform',
'show_receptacle_of_object_12_of_current_platform',
'show_receptacle_of_object_13_of_current_platform',
'show_receptacle_of_object_14_of_current_platform',
'show_receptacle_of_object_15_of_current_platform',
'show_receptacle_of_object_16_of_current_platform',
'show_receptacle_of_object_17_of_current_platform',
'show_receptacle_of_object_18_of_current_platform',
'show_receptacle_of_object_19_of_current_platform',

```

'show\_receptacle\_of\_object\_20\_of\_current\_platform',  
'show\_receptacle\_of\_object\_21\_of\_current\_platform',  
'show\_receptacle\_of\_object\_22\_of\_current\_platform',  
'show\_receptacle\_of\_object\_23\_of\_current\_platform',  
'show\_receptacle\_of\_object\_24\_of\_current\_platform',  
'show\_receptacle\_of\_object\_25\_of\_current\_platform',  
'show\_receptacle\_of\_object\_26\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'change\_view': Change the viewing angle to observe the scene from another perspective.

Choose one of the available actions.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 6/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 17]

Here are 4 images, named ['Idle\_6\_1\_out\_of\_4.png', 'Idle\_6\_2\_out\_of\_4.png', 'Idle\_6\_3\_out\_of\_4.png', 'Idle\_6\_4\_out\_of\_4.png'] respectively.

Their description is as follows: These images show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_kitchen\_counter\_1\_body\_platform\_1.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will be marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM\_response: pick\_object\_7\_of\_current\_platform  
-----

User Message:

You are currently at object\_kitchen\_counter\_1\_body\_platform\_1.

And you are holding kitchenware\_spice\_container.

Your available actions: -['go\_to\_object\_sofa\_10\_platform\_0',  
'go\_to\_object\_sofa\_10\_platform\_1', 'go\_to\_object\_sofa\_10\_platform\_2',  
'go\_to\_object\_table\_04\_13\_platform\_0', 'go\_to\_object\_chair\_04\_46\_platform\_0',  
'go\_to\_object\_chair\_04\_47\_platform\_0', 'go\_to\_object\_chair\_05\_7\_platform\_0',  
'go\_to\_object\_chair\_05\_8\_platform\_0', 'go\_to\_object\_stool\_02\_18\_platform\_0',  
'go\_to\_object\_stool\_02\_6\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_0',  
'go\_to\_object\_rack\_01\_75\_platform\_1', 'go\_to\_object\_rack\_01\_75\_platform\_2',

```

'go_to_object_kitchen_counter_1_body_platform_0',
'go_to_object_kitchen_counter_1_body_platform_1',
'go_to_object_fridge_0_body_platform_1', 'go_to_object_fridge_0_body_platform_3',
'go_to_object_fridge_0_body_platform_4', 'go_to_object_fridge_0_body_platform_5',
'go_to_object_fridge_0_body_platform_6', 'go_to_object_fridge_0_body_platform_7',
'go_to_object_bin_03_3_platform_0', 'go_to_object_wall_cabinet_01_4_platform_0',
'go_to_object_wall_cabinet_01_4_platform_1',
'go_to_object_wall_cabinet_01_4_platform_2',
'go_to_object_wall_cabinet_01_4_platform_3',
'go_to_object_wall_cabinet_01_4_platform_4',
'go_to_object_wall_cabinet_01_4_platform_5',
'go_to_object_wall_cabinet_01_4_platform_6', 'go_to_object_table_03_14_platform_1',
'go_to_object_table_01_48_platform_0', 'go_to_object_wall_cabinet_02_21_platform_0',
'go_to_object_wall_cabinet_02_21_platform_1',
'go_to_object_wall_cabinet_02_21_platform_2',
'go_to_object_wall_cabinet_02_21_platform_3',
'go_to_object_wall_cabinet_02_21_platform_4',
'go_to_object_wall_cabinet_02_21_platform_5',
'go_to_object_wall_cabinet_02_21_platform_6',
'go_to_object_chestOfDrawers_01_2_body_platform_0',
'go_to_object_chestOfDrawers_01_2_body_platform_1',
'go_to_object_chestOfDrawers_01_2_body_platform_2',
'go_to_object_chestOfDrawers_01_2_body_platform_3',
'go_to_object_chestOfDrawers_01_2_body_platform_4',
'go_to_object_chestOfDrawers_01_2_body_platform_5',
'go_to_object_chestOfDrawers_01_2_body_platform_6',
'go_to_object_table_02_40_platform_0', 'go_to_object_table_02_40_platform_1',
'go_to_object_table_02_40_platform_2', 'go_to_object_tvstand_88_platform_0',
'go_to_object_cabinet_3_body_platform_0', 'go_to_object_cabinet_3_body_platform_1'] -
['show_receptacle_of_object_1_of_current_platform',
'show_receptacle_of_object_2_of_current_platform',
'show_receptacle_of_object_3_of_current_platform',
'show_receptacle_of_object_4_of_current_platform',
'show_receptacle_of_object_5_of_current_platform',
'show_receptacle_of_object_6_of_current_platform',
'show_receptacle_of_object_7_of_current_platform',
'show_receptacle_of_object_8_of_current_platform',
'show_receptacle_of_object_9_of_current_platform',
'show_receptacle_of_object_10_of_current_platform',
'show_receptacle_of_object_11_of_current_platform',
'show_receptacle_of_object_12_of_current_platform',
'show_receptacle_of_object_13_of_current_platform',
'show_receptacle_of_object_14_of_current_platform',
'show_receptacle_of_object_15_of_current_platform',
'show_receptacle_of_object_16_of_current_platform',
'show_receptacle_of_object_17_of_current_platform',
'show_receptacle_of_object_18_of_current_platform',
'show_receptacle_of_object_19_of_current_platform',
'show_receptacle_of_object_20_of_current_platform',
'show_receptacle_of_object_21_of_current_platform',
'show_receptacle_of_object_22_of_current_platform',
'show_receptacle_of_object_23_of_current_platform',
'show_receptacle_of_object_24_of_current_platform',
'show_receptacle_of_object_25_of_current_platform'] - 'CALL_END': Call end (terminate
the task). - 'change_view': Change the viewing angle to observe the scene from another
perspective. - 'place_r': Place the object in an arbitrary place on the platform. -
'place_s_[(object_idx1,dir_idx1)...]': Place the object in the union area of regions with

```

indices (object\_idx1,region\_idx1)(object\_idx1's region\_idx1 space) etc. All available pairs are in [(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (5, 1), (6, 1), (6, 2), (7, 1), (8, 1), (9, 1), (10, 1), (10, 2), (10, 3), (11, 1), (11, 2), (11, 3), (11, 4), (11, 5), (11, 6), (11, 7), (12, 1), (12, 2), (13, 1), (13, 2), (13, 3), (13, 4), (13, 5), (15, 1), (15, 2), (15, 3), (16, 1), (17, 1), (18, 1), (18, 2), (18, 3), (18, 4), (19, 1), (19, 2), (19, 3), (19, 4), (19, 5), (20, 1), (20, 2), (20, 3), (21, 1), (22, 1), (22, 2), (22, 3), (22, 4), (22, 5), (22, 6), (22, 7), (22, 8), (23, 1), (23, 2), (24, 1), (24, 2), (24, 3), (25, 1), (25, 2), (25, 3), (25, 4), (25, 5)], and a valid choice should be a subset of it.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.

Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions: [front-left, front, front-right], [left, center, right], [rear-left, rear, rear-right]

Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 7/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 18]

Here are 4 images, named ['HoldingOccupiedPlatform7\_1\_out\_of\_4.png', 'HoldingOccupiedPlatform7\_2\_out\_of\_4.png', 'HoldingOccupiedPlatform7\_3\_out\_of\_4.png', 'HoldingOccupiedPlatform7\_4\_out\_of\_4.png'] respectively.

Their description is as follows: These image show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_kitchen\_counter\_1\_body\_platform\_1.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----

VLM Response: place\_r

User Message:

**The object has been placed successfully. You can now call end if you think you've finished the task correctly, or can also pick up the object again if you think the placement is not correct.**

You are currently at object\_kitchen\_counter\_1\_body\_platform\_1.

And you are holding nothing.

Your available action space:

```
- ['go_to_object_sofa_10_platform_0', 'go_to_object_sofa_10_platform_1',  
'go_to_object_sofa_10_platform_2', 'go_to_object_table_04_13_platform_0',  
'go_to_object_chair_04_46_platform_0', 'go_to_object_chair_04_47_platform_0',  
'go_to_object_chair_05_7_platform_0', 'go_to_object_chair_05_8_platform_0',  
'go_to_object_stool_02_18_platform_0', 'go_to_object_stool_02_6_platform_0',  
'go_to_object_rack_01_75_platform_0', 'go_to_object_rack_01_75_platform_1',  
'go_to_object_rack_01_75_platform_2',  
'go_to_object_kitchen_counter_1_body_platform_0',  
'go_to_object_kitchen_counter_1_body_platform_1',  
'go_to_object_fridge_0_body_platform_1', 'go_to_object_fridge_0_body_platform_3',  
'go_to_object_fridge_0_body_platform_4', 'go_to_object_fridge_0_body_platform_5',  
'go_to_object_fridge_0_body_platform_6', 'go_to_object_fridge_0_body_platform_7',  
'go_to_object_bin_03_3_platform_0', 'go_to_object_wall_cabinet_01_4_platform_0',  
'go_to_object_wall_cabinet_01_4_platform_1',  
'go_to_object_wall_cabinet_01_4_platform_2',  
'go_to_object_wall_cabinet_01_4_platform_3',  
'go_to_object_wall_cabinet_01_4_platform_4',  
'go_to_object_wall_cabinet_01_4_platform_5',  
'go_to_object_wall_cabinet_01_4_platform_6', 'go_to_object_table_03_14_platform_1',  
'go_to_object_table_01_48_platform_0', 'go_to_object_wall_cabinet_02_21_platform_0',  
'go_to_object_wall_cabinet_02_21_platform_1',  
'go_to_object_wall_cabinet_02_21_platform_2',  
'go_to_object_wall_cabinet_02_21_platform_3',  
'go_to_object_wall_cabinet_02_21_platform_4',  
'go_to_object_wall_cabinet_02_21_platform_5',  
'go_to_object_wall_cabinet_02_21_platform_6',  
'go_to_object_chestOfDrawers_01_2_body_platform_0',  
'go_to_object_chestOfDrawers_01_2_body_platform_1',  
'go_to_object_chestOfDrawers_01_2_body_platform_2',  
'go_to_object_chestOfDrawers_01_2_body_platform_3',  
'go_to_object_chestOfDrawers_01_2_body_platform_4',  
'go_to_object_chestOfDrawers_01_2_body_platform_5',  
'go_to_object_chestOfDrawers_01_2_body_platform_6',  
'go_to_object_table_02_40_platform_0', 'go_to_object_table_02_40_platform_1',  
'go_to_object_table_02_40_platform_2', 'go_to_object_tvstand_88_platform_0',  
'go_to_object_cabinet_3_body_platform_0', 'go_to_object_cabinet_3_body_platform_1'] -  
['pick_object_1_of_current_platform', 'pick_object_2_of_current_platform',  
'pick_object_3_of_current_platform', 'pick_object_4_of_current_platform',  
'pick_object_5_of_current_platform', 'pick_object_6_of_current_platform',  
'pick_object_7_of_current_platform', 'pick_object_8_of_current_platform',  
'pick_object_9_of_current_platform', 'pick_object_10_of_current_platform',  
'pick_object_11_of_current_platform', 'pick_object_12_of_current_platform',  
'pick_object_13_of_current_platform', 'pick_object_14_of_current_platform',  
'pick_object_15_of_current_platform', 'pick_object_16_of_current_platform',  
'pick_object_17_of_current_platform', 'pick_object_18_of_current_platform',  
'pick_object_19_of_current_platform', 'pick_object_20_of_current_platform',  
'pick_object_21_of_current_platform', 'pick_object_22_of_current_platform',  
'pick_object_23_of_current_platform', 'pick_object_24_of_current_platform',
```

'pick\_object\_25\_of\_current\_platform', 'pick\_object\_26\_of\_current\_platform'] -  
 ['show\_receptacle\_of\_object\_1\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_2\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_3\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_4\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_5\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_6\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_7\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_8\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_9\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_10\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_11\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_12\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_13\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_14\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_15\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_16\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_17\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_18\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_19\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_20\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_21\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_22\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_23\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_24\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_25\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_26\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'change\_view': Change the viewing angle to observe the scene from another perspective.

Choose one of the available actions.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 8/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 19]

Here are 4 images, named ['Idle\_8\_1\_out\_of\_4.png', 'Idle\_8\_2\_out\_of\_4.png', 'Idle\_8\_3\_out\_of\_4.png', 'Idle\_8\_4\_out\_of\_4.png'] respectively.

Their description is as follows: These images show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_kitchen\_counter\_1\_body\_platform\_1.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will be marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

VLM\_response: pick\_object\_1\_of\_current\_platform

User Message:

You are currently at object\_kitchen\_counter\_1\_body\_platform\_1.

And you are holding kitchenware\_white\_paper\_towel\_holder.

Your available actions: -['go\_to\_object\_sofa\_10\_platform\_0',

'go\_to\_object\_sofa\_10\_platform\_1', 'go\_to\_object\_sofa\_10\_platform\_2',

'go\_to\_object\_table\_04\_13\_platform\_0', 'go\_to\_object\_chair\_04\_46\_platform\_0',

'go\_to\_object\_chair\_04\_47\_platform\_0', 'go\_to\_object\_chair\_05\_7\_platform\_0',

'go\_to\_object\_chair\_05\_8\_platform\_0', 'go\_to\_object\_stool\_02\_18\_platform\_0',

'go\_to\_object\_stool\_02\_6\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_0',

'go\_to\_object\_rack\_01\_75\_platform\_1', 'go\_to\_object\_rack\_01\_75\_platform\_2',

'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_0',

'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_1',

'go\_to\_object\_fridge\_0\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_3',

'go\_to\_object\_fridge\_0\_body\_platform\_4', 'go\_to\_object\_fridge\_0\_body\_platform\_5',

'go\_to\_object\_fridge\_0\_body\_platform\_6', 'go\_to\_object\_fridge\_0\_body\_platform\_7',

'go\_to\_object\_bin\_03\_3\_platform\_0', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_0',

'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_1',

'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_2',

'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_3',

'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_4',

'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_5',

'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_6', 'go\_to\_object\_table\_03\_14\_platform\_1',

'go\_to\_object\_table\_01\_48\_platform\_0', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_0',

'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_1',

'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_2',

'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_3',

'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_4',

'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_5',

'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_6',

'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_0',

'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_1',

'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_2',

'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_3',

'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_4',

'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_5',

'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_6',

'go\_to\_object\_table\_02\_40\_platform\_0', 'go\_to\_object\_table\_02\_40\_platform\_1',

'go\_to\_object\_table\_02\_40\_platform\_2', 'go\_to\_object\_tvstand\_88\_platform\_0',

'go\_to\_object\_cabinet\_3\_body\_platform\_0', 'go\_to\_object\_cabinet\_3\_body\_platform\_1'] -

['show\_receptacle\_of\_object\_1\_of\_current\_platform',

'show\_receptacle\_of\_object\_2\_of\_current\_platform',

'show\_receptacle\_of\_object\_3\_of\_current\_platform',

'show\_receptacle\_of\_object\_4\_of\_current\_platform',

'show\_receptacle\_of\_object\_5\_of\_current\_platform',

'show\_receptacle\_of\_object\_6\_of\_current\_platform',

'show\_receptacle\_of\_object\_7\_of\_current\_platform',

'show\_receptacle\_of\_object\_8\_of\_current\_platform',



'show\_receptacle\_of\_object\_9\_of\_current\_platform',  
'show\_receptacle\_of\_object\_10\_of\_current\_platform',  
'show\_receptacle\_of\_object\_11\_of\_current\_platform',  
'show\_receptacle\_of\_object\_12\_of\_current\_platform',  
'show\_receptacle\_of\_object\_13\_of\_current\_platform',  
'show\_receptacle\_of\_object\_14\_of\_current\_platform',  
'show\_receptacle\_of\_object\_15\_of\_current\_platform',  
'show\_receptacle\_of\_object\_16\_of\_current\_platform',  
'show\_receptacle\_of\_object\_17\_of\_current\_platform',  
'show\_receptacle\_of\_object\_18\_of\_current\_platform',  
'show\_receptacle\_of\_object\_19\_of\_current\_platform',  
'show\_receptacle\_of\_object\_20\_of\_current\_platform',  
'show\_receptacle\_of\_object\_21\_of\_current\_platform',  
'show\_receptacle\_of\_object\_22\_of\_current\_platform',  
'show\_receptacle\_of\_object\_23\_of\_current\_platform',  
'show\_receptacle\_of\_object\_24\_of\_current\_platform',  
'show\_receptacle\_of\_object\_25\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'change\_view': Change the viewing angle to observe the scene from another perspective. - 'place\_r': Place the object in an arbitrary place on the platform. - 'place\_s\_[(object\_idx1,dir\_idx1)...]': Place the object in the union area of regions with indices (object\_idx1,region\_idx1)(object\_idx1's region\_idx1 space) etc. All available pairs are in [(1, 1), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (4, 1), (4, 2), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (7, 1), (7, 2), (7, 3), (8, 1), (9, 1), (10, 1), (10, 2), (10, 3), (11, 1), (11, 2), (11, 3), (11, 4), (11, 5), (12, 1), (12, 2), (12, 3), (13, 1), (13, 2), (13, 3), (13, 4), (14, 1), (14, 2), (14, 3), (14, 4), (14, 5), (14, 6), (14, 7), (14, 8), (15, 1), (15, 2), (16, 1), (16, 2), (16, 3), (17, 1), (17, 2), (17, 3), (17, 4), (17, 5), (18, 1), (18, 2), (18, 3), (18, 4), (19, 1), (19, 2), (19, 3), (19, 4), (19, 5), (19, 6), (20, 1), (20, 2), (20, 3), (20, 4), (21, 1), (21, 2), (21, 3), (21, 4), (22, 1), (22, 2), (22, 3), (22, 4), (23, 1), (23, 2), (23, 3), (24, 1), (24, 2), (25, 1)], and a valid choice should be a subset of it.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.

Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions: [front-left, front, front-right], [left, center, right], [rear-left, rear, rear-right]

Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.



4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 9/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 20]

Here are 4 images, named ['HoldingOccupiedPlatform9\_1\_out\_of\_4.png', 'HoldingOccupiedPlatform9\_2\_out\_of\_4.png', 'HoldingOccupiedPlatform9\_3\_out\_of\_4.png', 'HoldingOccupiedPlatform9\_4\_out\_of\_4.png'] respectively.

Their description is as follows: These image show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_kitchen\_counter\_1\_body\_platform\_1.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM Response: go\_to\_object\_table\_02\_40\_platform\_2  
-----

User Message:

You are currently at object\_table\_02\_40\_platform\_2.

And you are holding kitchenware\_white\_paper\_towel\_holder.

Your available action space:

-['go\_to\_object\_sofa\_10\_platform\_0', 'go\_to\_object\_sofa\_10\_platform\_1',  
'go\_to\_object\_sofa\_10\_platform\_2', 'go\_to\_object\_table\_04\_13\_platform\_0',  
'go\_to\_object\_chair\_04\_46\_platform\_0', 'go\_to\_object\_chair\_04\_47\_platform\_0',  
'go\_to\_object\_chair\_05\_7\_platform\_0', 'go\_to\_object\_chair\_05\_8\_platform\_0',  
'go\_to\_object\_stool\_02\_18\_platform\_0', 'go\_to\_object\_stool\_02\_6\_platform\_0',  
'go\_to\_object\_rack\_01\_75\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_1',  
'go\_to\_object\_rack\_01\_75\_platform\_2',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_0',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_1',  
'go\_to\_object\_fridge\_0\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_3',  
'go\_to\_object\_fridge\_0\_body\_platform\_4', 'go\_to\_object\_fridge\_0\_body\_platform\_5',  
'go\_to\_object\_fridge\_0\_body\_platform\_6', 'go\_to\_object\_fridge\_0\_body\_platform\_7',  
'go\_to\_object\_bin\_03\_3\_platform\_0', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_6', 'go\_to\_object\_table\_03\_14\_platform\_1',  
'go\_to\_object\_table\_01\_48\_platform\_0', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_6',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_0',

'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_1',  
 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_2',  
 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_3',  
 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_4',  
 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_5',  
 'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_6',  
 'go\_to\_object\_table\_02\_40\_platform\_0', 'go\_to\_object\_table\_02\_40\_platform\_1',  
 'go\_to\_object\_table\_02\_40\_platform\_2', 'go\_to\_object\_tvstand\_88\_platform\_0',  
 'go\_to\_object\_cabinet\_3\_body\_platform\_0', 'go\_to\_object\_cabinet\_3\_body\_platform\_1'] -  
 ['show\_receptacle\_of\_object\_1\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_2\_of\_current\_platform'] - 'CALL\_END': Call end (terminate  
 the task). - 'change\_view': Change the viewing angle to observe the scene from another  
 perspective. - 'place\_r': Place the object in an arbitrary place on the platform. -  
 'place\_s\_[(object\_idx1,dir\_idx1)...]': Place the object in the union area of regions with  
 indices (object\_idx1,region\_idx1)(object\_idx1's region\_idx1 space) etc. All available pairs  
 are in [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5),  
 (2, 6), (2, 7), (2, 8)], and a valid choice should be a subset of it.  
 All your current available actions are listed as above (all are without quotes and square  
 brackets). Please don't mix up it with the action space of previous steps.  
 Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative  
 to the robot's perspective and the heading of the object or platform you are interacting with.  
 The heading of a ground object is not fixed and depends on where the robot is standing and  
 facing.  
 Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided  
 into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to  
 you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions  
 proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right,"  
 and "front-right." The grid is conceptually numbered as follows, based on these directions:  
 [front-left, front, front-right],  
 [left, center, right],  
 [rear-left, rear, rear-right]  
 Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the  
 edges of the object's bounding box to the platform's boundaries. The resulting regions define  
 the object's 8 receptacles, and the direction is defined similarly. Here are some useful  
 suggestions for you to better finish the task:  
 1. Your extra actions do not affect the success of the task, you only need to keep your steps  
 within the total step limit.  
 2. When you want to place an object but cannot find enough space, you can try changing your  
 standing position to find the correct orientation, try combining multiple receptacles, or  
 remove objects that are blocking your way first(although you generally do not need to do  
 this). Specifically, for the task asking you put object to empty platforms, try combining  
 adjacent receptacles may be very useful.  
 3. For tasks that involve placing objects near other objects, please remember to get a focused  
 image of the target object and examine it carefully. This is important because the receptacles  
 may not be intuitive.  
 4. Important: Don't output anything other than the action you want to take. You will only  
 receive the same hint informing you your invalid action. If you make mistakes, just start  
 output correct actions from the next prompt without any apologize.  
 Current task:  
 Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and  
 lighting\_blue\_base\_table\_lamp  
 Steps used: 10/20. You can only take at most 20 steps, so hurry if you've almost used all of  
 them!  
 [\*See Fig. 21]  
 Here are 2 images, named ['HoldingOccupiedPlatform10\_1\_out\_of\_2.png',  
 'HoldingOccupiedPlatform10\_2\_out\_of\_2.png'] respectively.

Their description are as follows: These image show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_table\_02\_40\_platform\_2.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM Response: show\_receptacle\_of\_object\_1\_of\_current\_platform  
-----

User Message:

[\*See Fig. 22 *ShowReceptacle\_11.png*]

Here are 1 images, named ['ShowReceptacle\_11.png'] respectively.

Their description is as follows: This image shows the available receptacles around object 1 on the platform, you see this because you've taken "show\_receptacle" action previously. The numbers from 1 to 8 indicate the available receptacles. Not all numbers from 1 to 8 may be present, as some receptacles might be too small or blocked. To use the numbers shown in the image to specify where you want to place the object you are holding, you can include (1, receptacles\_idx (receptacles\_idx in [1, 8]) in the output action.

You are currently at object\_table\_02\_40\_platform\_2.

And you are holding kitchenware\_white\_paper\_towel\_holder.

Your available action space:

```
-['go_to_object_sofa_10_platform_0', 'go_to_object_sofa_10_platform_1',  
'go_to_object_sofa_10_platform_2', 'go_to_object_table_04_13_platform_0',  
'go_to_object_chair_04_46_platform_0', 'go_to_object_chair_04_47_platform_0',  
'go_to_object_chair_05_7_platform_0', 'go_to_object_chair_05_8_platform_0',  
'go_to_object_stool_02_18_platform_0', 'go_to_object_stool_02_6_platform_0',  
'go_to_object_rack_01_75_platform_0', 'go_to_object_rack_01_75_platform_1',  
'go_to_object_rack_01_75_platform_2',  
'go_to_object_kitchen_counter_1_body_platform_0',  
'go_to_object_kitchen_counter_1_body_platform_1',  
'go_to_object_fridge_0_body_platform_1', 'go_to_object_fridge_0_body_platform_3',  
'go_to_object_fridge_0_body_platform_4', 'go_to_object_fridge_0_body_platform_5',  
'go_to_object_fridge_0_body_platform_6', 'go_to_object_fridge_0_body_platform_7',  
'go_to_object_bin_03_3_platform_0', 'go_to_object_wall_cabinet_01_4_platform_0',  
'go_to_object_wall_cabinet_01_4_platform_1',  
'go_to_object_wall_cabinet_01_4_platform_2',  
'go_to_object_wall_cabinet_01_4_platform_3',  
'go_to_object_wall_cabinet_01_4_platform_4',  
'go_to_object_wall_cabinet_01_4_platform_5',  
'go_to_object_wall_cabinet_01_4_platform_6', 'go_to_object_table_03_14_platform_1',  
'go_to_object_table_01_48_platform_0', 'go_to_object_wall_cabinet_02_21_platform_0',  
'go_to_object_wall_cabinet_02_21_platform_1',  
'go_to_object_wall_cabinet_02_21_platform_2',  
'go_to_object_wall_cabinet_02_21_platform_3',  
'go_to_object_wall_cabinet_02_21_platform_4',  
'go_to_object_wall_cabinet_02_21_platform_5',  
'go_to_object_wall_cabinet_02_21_platform_6',  
'go_to_object_chestOfDrawers_01_2_body_platform_0',  
'go_to_object_chestOfDrawers_01_2_body_platform_1',  
'go_to_object_chestOfDrawers_01_2_body_platform_2',  
'go_to_object_chestOfDrawers_01_2_body_platform_3',  
'go_to_object_chestOfDrawers_01_2_body_platform_4',  
'go_to_object_chestOfDrawers_01_2_body_platform_5',
```

'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_6',  
 'go\_to\_object\_table\_02\_40\_platform\_0', 'go\_to\_object\_table\_02\_40\_platform\_1',  
 'go\_to\_object\_table\_02\_40\_platform\_2', 'go\_to\_object\_tvstand\_88\_platform\_0',  
 'go\_to\_object\_cabinet\_3\_body\_platform\_0', 'go\_to\_object\_cabinet\_3\_body\_platform\_1'] -  
 ['show\_receptacle\_of\_object\_1\_of\_current\_platform',  
 'show\_receptacle\_of\_object\_2\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'change\_view': Change the viewing angle to observe the scene from another perspective. - 'place\_r': Place the object in an arbitrary place on the platform. - 'place\_s\_[(object\_idx1,dir\_idx1)...]': Place the object in the union area of regions with indices (object\_idx1,region\_idx1)(object\_idx1's region\_idx1 space) etc. All available pairs are in [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8)], and a valid choice should be a subset of it.  
 All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.  
 Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.  
 Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions:  
 [front-left, front, front-right],  
 [left, center, right],  
 [rear-left, rear, rear-right]  
 Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Here are some useful suggestions for you to better finish the task:  
 1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.  
 2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.  
 3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.  
 4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.  
 Current task:  
 Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp  
 Steps used: 11/20. You can only take at most 20 steps, so hurry if you've almost used all of them!  
 [\*See Fig. 22 *HoldingOccupiedPlatform11\_1\_out\_of\_2.png,HoldingOccupiedPlatform11\_2\_out\_of\_2.png*]  
 Here are 2 images, named ['HoldingOccupiedPlatform11\_1\_out\_of\_2.png', 'HoldingOccupiedPlatform11\_2\_out\_of\_2.png'] respectively.  
 Their description are as follows: These images show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_table\_02\_40\_platform\_2.  
 If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform,

guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will be marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM Response: [show\\_receptacle\\_of\\_object\\_2\\_of\\_current\\_platform](#)  
-----

User Message:

[\*See Fig. 23 *ShowReceptacle\_12.png*]

Here are 1 images, named ['ShowReceptacle\_12.png'] respectively.

Their description is as follows: This image shows the available receptacles around object 1 on the platform, you see this because you've taken "show\_receptacle" action previously. The numbers from 1 to 8 indicate the available receptacles. Not all numbers from 1 to 8 may be present, as some receptacles might be too small or blocked. To use the numbers shown in the image to specify where you want to place the object you are holding, you can include (1, receptacles\_idx (receptacles\_idx in [1, 8]) in the output action.

You are currently at object\_table\_02\_40\_platform\_2.

And you are holding kitchenware\_white\_paper\_towel\_holder.

Your available action space:

```
-['go_to_object_sofa_10_platform_0', 'go_to_object_sofa_10_platform_1',  
'go_to_object_sofa_10_platform_2', 'go_to_object_table_04_13_platform_0',  
'go_to_object_chair_04_46_platform_0', 'go_to_object_chair_04_47_platform_0',  
'go_to_object_chair_05_7_platform_0', 'go_to_object_chair_05_8_platform_0',  
'go_to_object_stool_02_18_platform_0', 'go_to_object_stool_02_6_platform_0',  
'go_to_object_rack_01_75_platform_0', 'go_to_object_rack_01_75_platform_1',  
'go_to_object_rack_01_75_platform_2',  
'go_to_object_kitchen_counter_1_body_platform_0',  
'go_to_object_kitchen_counter_1_body_platform_1',  
'go_to_object_fridge_0_body_platform_1', 'go_to_object_fridge_0_body_platform_3',  
'go_to_object_fridge_0_body_platform_4', 'go_to_object_fridge_0_body_platform_5',  
'go_to_object_fridge_0_body_platform_6', 'go_to_object_fridge_0_body_platform_7',  
'go_to_object_bin_03_3_platform_0', 'go_to_object_wall_cabinet_01_4_platform_0',  
'go_to_object_wall_cabinet_01_4_platform_1',  
'go_to_object_wall_cabinet_01_4_platform_2',  
'go_to_object_wall_cabinet_01_4_platform_3',  
'go_to_object_wall_cabinet_01_4_platform_4',  
'go_to_object_wall_cabinet_01_4_platform_5',  
'go_to_object_wall_cabinet_01_4_platform_6', 'go_to_object_table_03_14_platform_1',  
'go_to_object_table_01_48_platform_0', 'go_to_object_wall_cabinet_02_21_platform_0',  
'go_to_object_wall_cabinet_02_21_platform_1',  
'go_to_object_wall_cabinet_02_21_platform_2',  
'go_to_object_wall_cabinet_02_21_platform_3',  
'go_to_object_wall_cabinet_02_21_platform_4',  
'go_to_object_wall_cabinet_02_21_platform_5',  
'go_to_object_wall_cabinet_02_21_platform_6',  
'go_to_object_chestOfDrawers_01_2_body_platform_0',  
'go_to_object_chestOfDrawers_01_2_body_platform_1',  
'go_to_object_chestOfDrawers_01_2_body_platform_2',  
'go_to_object_chestOfDrawers_01_2_body_platform_3',  
'go_to_object_chestOfDrawers_01_2_body_platform_4',  
'go_to_object_chestOfDrawers_01_2_body_platform_5',  
'go_to_object_chestOfDrawers_01_2_body_platform_6',  
'go_to_object_table_02_40_platform_0', 'go_to_object_table_02_40_platform_1',  
'go_to_object_table_02_40_platform_2', 'go_to_object_tvstand_88_platform_0',  
'go_to_object_cabinet_3_body_platform_0', 'go_to_object_cabinet_3_body_platform_1'] -  
['show_receptacle_of_object_1_of_current_platform',
```

'show\_receptacle\_of\_object\_2\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'change\_view': Change the viewing angle to observe the scene from another perspective. - 'place\_r': Place the object in an arbitrary place on the platform. - 'place\_s\_[(object\_idx1,dir\_idx1)...]': Place the object in the union area of regions with indices (object\_idx1,region\_idx1)(object\_idx1's region\_idx1 space) etc. All available pairs are in [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8)], and a valid choice should be a subset of it.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.

Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions: [front-left, front, front-right],

[left, center, right],

[rear-left, rear, rear-right]

Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 12/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 23 *HoldingOccupiedPlatform12\_1\_out\_of\_2.png*,

*HoldingOccupiedPlatform12\_2\_out\_of\_2.png*]

Here are 2 images, named ['HoldingOccupiedPlatform12\_1\_out\_of\_2.png', 'HoldingOccupiedPlatform12\_2\_out\_of\_2.png'] respectively.

Their description are as follows: These images show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_table\_02\_40\_platform\_2.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----



VLM Response: place\_s\_[(1,7),(1,8),(2,3),(2,4)]

-----  
User Message:

**The object has been placed successfully. You can now call end if you think you've finished the task correctly, or can also pick up the object again if you think the placement is not correct.**

You are currently at object\_table\_02\_40\_platform\_2.

And you are holding nothing.

Your available action space:

- ['go\_to\_object\_sofa\_10\_platform\_0', 'go\_to\_object\_sofa\_10\_platform\_1',  
'go\_to\_object\_sofa\_10\_platform\_2', 'go\_to\_object\_table\_04\_13\_platform\_0',  
'go\_to\_object\_chair\_04\_46\_platform\_0', 'go\_to\_object\_chair\_04\_47\_platform\_0',  
'go\_to\_object\_chair\_05\_7\_platform\_0', 'go\_to\_object\_chair\_05\_8\_platform\_0',  
'go\_to\_object\_stool\_02\_18\_platform\_0', 'go\_to\_object\_stool\_02\_6\_platform\_0',  
'go\_to\_object\_rack\_01\_75\_platform\_0', 'go\_to\_object\_rack\_01\_75\_platform\_1',  
'go\_to\_object\_rack\_01\_75\_platform\_2',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_0',  
'go\_to\_object\_kitchen\_counter\_1\_body\_platform\_1',  
'go\_to\_object\_fridge\_0\_body\_platform\_1', 'go\_to\_object\_fridge\_0\_body\_platform\_3',  
'go\_to\_object\_fridge\_0\_body\_platform\_4', 'go\_to\_object\_fridge\_0\_body\_platform\_5',  
'go\_to\_object\_fridge\_0\_body\_platform\_6', 'go\_to\_object\_fridge\_0\_body\_platform\_7',  
'go\_to\_object\_bin\_03\_3\_platform\_0', 'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_01\_4\_platform\_6', 'go\_to\_object\_table\_03\_14\_platform\_1',  
'go\_to\_object\_table\_01\_48\_platform\_0', 'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_0',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_1',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_2',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_3',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_4',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_5',  
'go\_to\_object\_wall\_cabinet\_02\_21\_platform\_6',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_0',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_1',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_2',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_3',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_4',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_5',  
'go\_to\_object\_chestOfDrawers\_01\_2\_body\_platform\_6',  
'go\_to\_object\_table\_02\_40\_platform\_0', 'go\_to\_object\_table\_02\_40\_platform\_1',  
'go\_to\_object\_table\_02\_40\_platform\_2', 'go\_to\_object\_tvstand\_88\_platform\_0',  
'go\_to\_object\_cabinet\_3\_body\_platform\_0', 'go\_to\_object\_cabinet\_3\_body\_platform\_1'] -  
['pick\_object\_1\_of\_current\_platform', 'pick\_object\_2\_of\_current\_platform',  
'pick\_object\_3\_of\_current\_platform'] -  
['show\_receptacle\_of\_object\_1\_of\_current\_platform',  
'show\_receptacle\_of\_object\_2\_of\_current\_platform',  
'show\_receptacle\_of\_object\_3\_of\_current\_platform'] - 'CALL\_END': Call end (terminate  
the task). - 'change\_view': Change the viewing angle to observe the scene from another  
perspective.

Choose one of the available actions.

All your current available actions are listed as above (all are without quotes and square  
brackets). Please don't mix up it with the action space of previous steps.

Here are some useful suggestions for you to better finish the task:



1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 13/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 24]

Here are 2 images, named ['Idle\_13\_1\_out\_of\_2.png', 'Idle\_13\_2\_out\_of\_2.png'] respectively.

Their description is as follows: These image show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_table\_02\_40\_platform\_2.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM Response: CALL\_END

-----  
(call\_ended)

Task\_result:

Target object placed between two objects.

Task\_status:

Success

665 **D.2.2 Failed Episode**

666 The following episode shows how Gemini-2.5-pro failed in attempting a task in AI2THOR. It picked  
667 up the object and went to the destination correctly, but it failed to recognize the correct receptacle  
668 region.

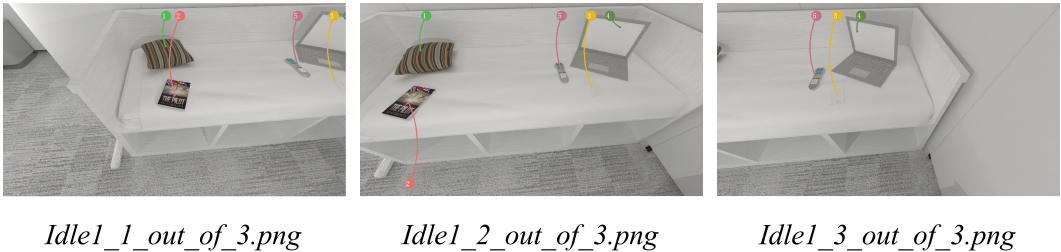


Figure 25: Images Given after Step 1 of the Failed Episode.

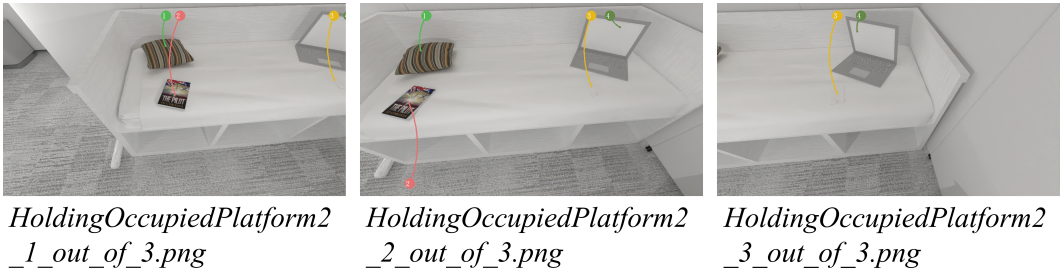


Figure 26: Images Given after Step 2 of the Failed Episode.

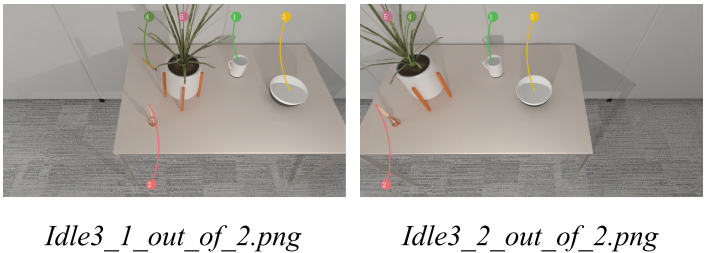


Figure 27: Images Given after Step 3 of the Failed Episode.

**Unsuccessful Episode by Gemini-2.5-pro**

System Message: You are a household assistant robot equipped with a vision system and manipulation capabilities. You need to complete a task in a 3D environment, mainly involving moving, picking up, and placing objects. Your goal is to move an object from one place to another correctly according to instructions. Please strictly follow the given Action Space for operations.

At the beginning of each task, you will be informed of a specific task to complete. Throughout the task execution, you will transition between multiple states and have access to a set of possible actions (action space).

You may receive one of the following task types involving one object and another object or a platform. (an object/another object/a platform are the name of objects or platforms, dir is one of the 8 directions):

669



Figure 28: Images Given after Step 4 of the Failed Episode.

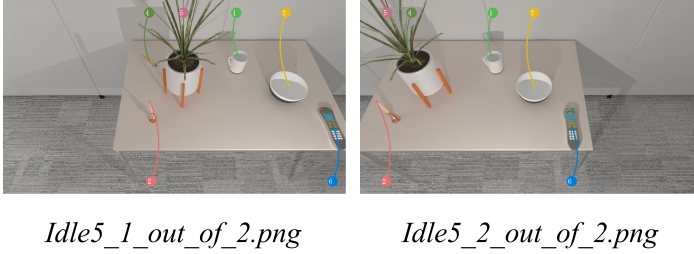


Figure 29: Images Given After Step 5 of the Failed Episode.

- move an object to a platform.
  - move an object to the dir of a platform.
  - move an object around another object.
  - move an object to the dir of another object.
  - move an object between another object and yet another object.
- Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.
- Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions:
- [front-left, front, front-right],
  - [left, center, right],
  - [rear-left, rear, rear-right]
- Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly.
- Task success criteria:
- move an object to a platform: The task is considered successful only when an object is placed anywhere on the specified platform.
  - move an object to the dir of a platform: The task is considered successful only when an object is placed on the platform and the geometric center of the placement is in the dir direction relative to the platform.
  - move an object around another object: The task is considered successful only when an object is placed in any of the receptacles surrounding another object.
  - move an object to the dir of another object: The task is considered successful only when an object is placed in the receptacles surrounding another object and overlaps with the dir direction.
  - move an object between another object and yet another object: The task is considered successful only when an object is placed such that it intersects with receptacles in a pair of opposite directions between another object and yet another object.

Your action space, includes walking to platforms, picking up items, putting items down, rotating direction, getting information about directions around an object; `CALL_END`, placing an item in a specific direction on a platform, in a direction relative to another object, or placing it anywhere. Whenever you're about to choose an action, I will provide your current action space. If you choose an action that cannot be performed, such as trying to pick up an object repeatedly or attempting to place an object in a space that's too small, you'll receive a notification.

Picking up an item: Corresponds to the command "pick\_object\_object number\_of\_current\_platform".

Walking to a platform: Corresponds to the command "go\_to\_platform name".

Rotating direction: Corresponds to the command "change\_view".

Getting an image of an object: Corresponds to the command "show\_receptacle\_of\_object\_object\_number\_of\_current\_platform".

Putting down your holding item on empty platform: Corresponds to the command "place\_s\_[region\_idx1,region\_idx2...]" where dir\_idx1, dir\_idx2 are NUMBERS, indicating the index of the receptacles; Putting down your item on an occupied platform: Corresponds to the command "place\_s\_[(object\_idx1,dir\_idx1)...]", where object\_idx1, dir\_idx1 are NUMBERS, indicating the object and its corresponding receptacles. For example, "place\_s\_[(1,2), (2,5)]" means putting down your item at the union of object 1's 2nd receptacles and object 2's 5th receptacles. Note that for placement action, the system will attempt to find a valid placement area that intersects with all the regions you selected. If it fails, it will place the object anywhere within your chosen space and you'll receive a notification.

You may be shown various types of images:

0) At the beginning of a task, if the task involves objects on a platform that contains other objects of the same type (e.g., the object is a book on a bookshelf with many books), you will receive a notification and images of these objects to help you distinguish them.

1) Whenever you move to a platform, you will see an image of the platform from your current perspective (or multiple images for longer platforms, indicating you can move horizontally to better observe the platform). If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects. If there are objects on the platform, guiding lines will indicate the object numbers.

2) If there are objects on a platform, you can use "show\_receptacle\_of\_object\_x\_of\_current\_platform" to get a focused image of that object with numbered receptacles around it. This allows you to see the available receptacles around the object.

Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first (although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move Remote to Bowl's rear-left receptacles Initially, Remote is on object\_hemnes\_day\_bed\_platform\_5, and Bowl is on object\_dining\_table\_mella\_platform\_0. Steps used: 0/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

Your available action space:

- 'CALL\_END': Call end (terminate the task). -  
 ['go\_to\_object\_hemnes\_day\_bed\_platform\_5', 'go\_to\_object\_bin\_6\_platform\_0',  
 'go\_to\_object\_Side\_Table\_317\_1\_platform\_1',  
 'go\_to\_object\_side\_table\_strind\_platform\_0',  
 'go\_to\_object\_side\_table\_strind\_platform\_1',  
 'go\_to\_object\_Side\_Table\_203\_1\_platform\_0',  
 'go\_to\_object\_Side\_Table\_203\_1\_platform\_1', 'go\_to\_object\_chair\_antnas\_platform\_0',  
 'go\_to\_object\_dresser\_birkeland\_platform\_0',  
 'go\_to\_object\_dresser\_birkeland\_platform\_1',  
 'go\_to\_object\_dresser\_birkeland\_platform\_2',  
 'go\_to\_object\_dresser\_birkeland\_platform\_3',  
 'go\_to\_object\_dresser\_birkeland\_platform\_4',  
 'go\_to\_object\_dresser\_birkeland\_platform\_5',  
 'go\_to\_object\_dresser\_birkeland\_platform\_6', 'go\_to\_object\_Basketball\_1\_platform\_0',  
 'go\_to\_object\_dining\_table\_mella\_platform\_0',  
 'go\_to\_object\_Floor\_Lamp\_13\_platform\_0',  
 'go\_to\_object\_armchair\_emmabo\_platform\_0', 'go\_to\_object\_BaseballBat\_2\_platform\_0'].

Please choose one of the available actions.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

-----  
 VLM Response: go\_to\_object\_hemnes\_day\_bed\_platform\_5  
 -----

User Message:

You are currently at object\_hemnes\_day\_bed\_platform\_5.

And you are holding nothing.

Your available actions:

- ['go\_to\_object\_hemnes\_day\_bed\_platform\_5', 'go\_to\_object\_bin\_6\_platform\_0',  
 'go\_to\_object\_Side\_Table\_317\_1\_platform\_1',  
 'go\_to\_object\_side\_table\_strind\_platform\_0',  
 'go\_to\_object\_side\_table\_strind\_platform\_1',  
 'go\_to\_object\_Side\_Table\_203\_1\_platform\_0',  
 'go\_to\_object\_Side\_Table\_203\_1\_platform\_1', 'go\_to\_object\_chair\_antnas\_platform\_0',  
 'go\_to\_object\_dresser\_birkeland\_platform\_0',  
 'go\_to\_object\_dresser\_birkeland\_platform\_1',  
 'go\_to\_object\_dresser\_birkeland\_platform\_2',  
 'go\_to\_object\_dresser\_birkeland\_platform\_3',  
 'go\_to\_object\_dresser\_birkeland\_platform\_4',  
 'go\_to\_object\_dresser\_birkeland\_platform\_5',  
 'go\_to\_object\_dresser\_birkeland\_platform\_6', 'go\_to\_object\_Basketball\_1\_platform\_0',  
 'go\_to\_object\_dining\_table\_mella\_platform\_0',  
 'go\_to\_object\_Floor\_Lamp\_13\_platform\_0',  
 'go\_to\_object\_armchair\_emmabo\_platform\_0', 'go\_to\_object\_BaseballBat\_2\_platform\_0'] -  
 ['pick\_object\_1\_of\_current\_platform', 'pick\_object\_2\_of\_current\_platform',  
 'pick\_object\_3\_of\_current\_platform', 'pick\_object\_4\_of\_current\_platform',  
 'pick\_object\_5\_of\_current\_platform'] -  
 ['show\_receptacles\_of\_object\_1\_of\_current\_platform',  
 'show\_receptacles\_of\_object\_2\_of\_current\_platform',  
 'show\_receptacles\_of\_object\_3\_of\_current\_platform',  
 'show\_receptacles\_of\_object\_4\_of\_current\_platform',  
 'show\_receptacles\_of\_object\_5\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'rotate\_observation\_view\_of\_current\_platform': Change the viewing angle to observe the scene from another perspective.

Choose one of the available actions.

All your current available actions are listed as above (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the receptacles may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

Current task:

Move kitchenware\_white\_paper\_towel\_holder between lighting\_blue\_base\_table\_lamp and lighting\_blue\_base\_table\_lamp

Steps used: 1/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

[\*See Fig. 25]

Here are 3 images, named ['Idle\_1\_1\_out\_of\_3.png', 'Idle\_1\_2\_out\_of\_3.png', 'Idle\_1\_3\_out\_of\_3.png'] respectively. Their description is as follows: These images show the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_hemnes\_day\_bed\_platform\_5. If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM Response: pick\_object\_5\_of\_current\_platform  
-----

You are currently at object\_hemnes\_day\_bed\_platform\_5.

And you are holding Remote.

Your available action space:

Your available actions: -['go\_to\_object\_hemnes\_day\_bed\_platform\_5',  
'go\_to\_object\_bin\_6\_platform\_0', 'go\_to\_object\_Side\_Table\_317\_1\_platform\_1',  
'go\_to\_object\_side\_table\_strind\_platform\_0',  
'go\_to\_object\_side\_table\_strind\_platform\_1',  
'go\_to\_object\_Side\_Table\_203\_1\_platform\_0',  
'go\_to\_object\_Side\_Table\_203\_1\_platform\_1', 'go\_to\_object\_chair\_antnas\_platform\_0',  
'go\_to\_object\_dresser\_birkeland\_platform\_0',  
'go\_to\_object\_dresser\_birkeland\_platform\_1',  
'go\_to\_object\_dresser\_birkeland\_platform\_2',  
'go\_to\_object\_dresser\_birkeland\_platform\_3',  
'go\_to\_object\_dresser\_birkeland\_platform\_4',  
'go\_to\_object\_dresser\_birkeland\_platform\_5',  
'go\_to\_object\_dresser\_birkeland\_platform\_6', 'go\_to\_object\_Basketball\_1\_platform\_0',  
'go\_to\_object\_dining\_table\_mella\_platform\_0',  
'go\_to\_object\_Floor\_Lamp\_13\_platform\_0',  
'go\_to\_object\_armchair\_emmabo\_platform\_0', 'go\_to\_object\_BaseballBat\_2\_platform\_0']  
- ['show\_receptacles\_of\_object\_1\_of\_current\_platform',  
'show\_receptacles\_of\_object\_2\_of\_current\_platform',  
'show\_receptacles\_of\_object\_3\_of\_current\_platform',  
'show\_receptacles\_of\_object\_4\_of\_current\_platform'] - 'CALL\_END': Call end (terminate the task). - 'rotate\_observation\_view\_of\_current\_platform': Change the viewing angle to



observe the scene from another perspective. - 'place\_at\_anywhere': Place the object in an arbitrary place on the platform. - 'place\_at\_receptacles\_[(object\_idx1,dir\_idx1)...]': Place the object in the union area of regions with indices (object\_idx1,region\_idx1)(object\_idx1's region\_idx1 space) etc. All available pairs are in [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7)], and a valid choice should be a subset of it.

All your current available actions are listed as follows (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Current task:

Move Remote to Bowl's rear-left receptacles

Steps used: 2/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.

Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions: [front-left, front, front-right], [left, center, right], [rear-left, rear, rear-right]

Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the regions may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

[\*See Fig. 26] Here are 3 images, named ['HoldingOccupiedPlatformState\_2\_1\_out\_of\_3.png', 'HoldingOccupiedPlatformState\_2\_2\_out\_of\_3.png', 'HoldingOccupiedPlatformState\_2\_3\_out\_of\_3.png'] respectively.

Their description is as follows: This image shows the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_hemnes\_day\_bed\_platform\_5.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

VLM Response: go\_to\_object\_dining\_table\_mella\_platform\_0

You are currently at object\_dining\_table\_mella\_platform\_0.



And you are holding Remote.

Your available action space:

```
['go_to_object_hemnes_day_bed_platform_5', 'go_to_object_bin_6_platform_0',  
'go_to_object_Side_Table_317_1_platform_1',  
'go_to_object_side_table_strind_platform_0',  
'go_to_object_side_table_strind_platform_1',  
'go_to_object_Side_Table_203_1_platform_0',  
'go_to_object_Side_Table_203_1_platform_1', 'go_to_object_chair_antnas_platform_0',  
'go_to_object_dresser_birkeland_platform_0',  
'go_to_object_dresser_birkeland_platform_1',  
'go_to_object_dresser_birkeland_platform_2',  
'go_to_object_dresser_birkeland_platform_3',  
'go_to_object_dresser_birkeland_platform_4',  
'go_to_object_dresser_birkeland_platform_5',  
'go_to_object_dresser_birkeland_platform_6', 'go_to_object_Basketball_1_platform_0',  
'go_to_object_dining_table_mella_platform_0',  
'go_to_object_Floor_Lamp_13_platform_0',  
'go_to_object_armchair_emmabo_platform_0', 'go_to_object_BaseballBat_2_platform_0']  
- ['show_receptacles_of_object_1_of_current_platform',  
'show_receptacles_of_object_2_of_current_platform',  
'show_receptacles_of_object_3_of_current_platform',  
'show_receptacles_of_object_4_of_current_platform',  
'show_receptacles_of_object_5_of_current_platform'] - 'CALL_END': Call end (terminate  
the task). - 'rotate_observation_view_of_current_platform': Change the viewing angle to  
observe the scene from another perspective. - 'place_at_anywhere': Place the object in an  
arbitrary place on the platform. - 'place_at_receptacles_[(object_idx1,dir_idx1)...]': Place the  
object in the union area of regions with indices (object_idx1,region_idx1)(object_idx1's  
region_idx1 space) etc. All available pairs are in [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1,  
7), (1, 8), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (3,  
8), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)], and a  
valid choice should be a subset of it.
```

All your current available actions are listed as follows (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Current task:

Move Remote to Bowl's rear-left receptacles

Steps used: 3/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.

Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions: [front-left, front, front-right], [left, center, right], [rear-left, rear, rear-right]

Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do

this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.

3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the regions may not be intuitive.

4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

[\*See Fig. 27]

Here are 2 images, named ['HoldingOccupiedPlatform\_3\_1\_out\_of\_2.png', 'HoldingOccupiedPlatform\_3\_2\_out\_of\_2.png'] respectively.

Their description is as follows: This image shows the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_dining\_table\_mella\_platform\_0.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM Response: show\_receptacle\_of\_object\_5\_of\_current\_platform  
-----

[\* See Fig. 28 ShowReceptacle\_4.png]

Here are 1 images, named ['ShowReceptacle\_4.png'] respectively.

Their description is as follows: This image shows the available receptacles around object 3 on the platform, you see this because you've taken "show\_receptacles" action previously. The numbers from 1 to 8 indicate the available receptacles. Not all numbers from 1 to 8 may be present, as some receptacles might be too small or blocked. To use the numbers shown in the image to specify where you want to place the object you are holding, you can include (3, receptacles\_idx (receptacles\_idx in [1, 8]) in the output action.

You are currently at  
object\_dining\_table\_mella\_platform\_0.

And you are holding Remote.

Your available action space:

```
-['go_to_object_hemnes_day_bed_platform_5', 'go_to_object_bin_6_platform_0',  
'go_to_object_Side_Table_317_1_platform_1',  
'go_to_object_side_table_strind_platform_0',  
'go_to_object_side_table_strind_platform_1',  
'go_to_object_Side_Table_203_1_platform_0',  
'go_to_object_Side_Table_203_1_platform_1', 'go_to_object_chair_antnas_platform_0',  
'go_to_object_dresser_birkeland_platform_0',  
'go_to_object_dresser_birkeland_platform_1',  
'go_to_object_dresser_birkeland_platform_2',  
'go_to_object_dresser_birkeland_platform_3',  
'go_to_object_dresser_birkeland_platform_4',  
'go_to_object_dresser_birkeland_platform_5',  
'go_to_object_dresser_birkeland_platform_6', 'go_to_object_Basketball_1_platform_0',  
'go_to_object_dining_table_mella_platform_0',  
'go_to_object_Floor_Lamp_13_platform_0',  
'go_to_object_armchair_emmabo_platform_0', 'go_to_object_BaseballBat_2_platform_0']  
-['show_receptacles_of_object_1_of_current_platform',  
'show_receptacles_of_object_2_of_current_platform',  
'show_receptacles_of_object_3_of_current_platform',  
'show_receptacles_of_object_4_of_current_platform',  
'show_receptacles_of_object_5_of_current_platform'] - 'CALL_END': Call end (terminate
```

the task). - 'rotate\_observation\_view\_of\_current\_platform': Change the viewing angle to observe the scene from another perspective. - 'place\_at\_anywhere': Place the object in an arbitrary place on the platform. - 'place\_at\_receptacles\_[(object\_idx1,dir\_idx1)...]': Place the object in the union area of regions with indices (object\_idx1,region\_idx1)(object\_idx1's region\_idx1 space) etc. All available pairs are in [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (3, 8), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5)], and a valid choice should be a subset of it.

All your current available actions are listed as follows (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Current task:

Move Remote to Bowl's rear-left receptacles

Steps used: 4/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

Understanding directions: Directions like 'front', 'rear', 'left', and 'right' are defined relative to the robot's perspective and the heading of the object or platform you are interacting with. The heading of a ground object is not fixed and depends on where the robot is standing and facing.

Empty Surfaces (Platforms): When interacting with an empty platform, imagine it divided into a 3x3 grid. The direction aligned with the heading (also, the FARTHEST middle space to you, NOT THE NEAREST MIDDLE SPACE) is labeled "front," with the remaining regions proceeding counterclockwise as "front-left," "left," "rear-left," "rear," "rear-right," "right," and "front-right." The grid is conceptually numbered as follows, based on these directions: [front-left, front, front-right], [left, center, right], [rear-left, rear, rear-right]

Occupied Surfaces (Objects on Platforms): For objects on platforms, imagine extending the edges of the object's bounding box to the platform's boundaries. The resulting regions define the object's 8 receptacles, and the direction is defined similarly. Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the regions may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

[\* See Fig. 28 *HoldingOccupiedPlatform4\_1\_out\_of\_2.png*

*form\_4\_1\_out\_of\_2.png,HoldingOccupiedPlatform\_4\_2\_out\_of\_2.png]*

Here are 2 images, named ['HoldingOccupiedPlatform4\_1\_out\_of\_2.png', 'HoldingOccupiedPlatform4\_2\_out\_of\_2.png'] respectively.

Their description is as follows: This image shows the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_dining\_table\_mella\_platform\_0.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will be marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

VLM\_response: place\_s\_[(3,8)]

**The object has been placed successfully. You can now call end if you think you've finished the task correctly, or can also pick up the object again if you think the placement is not correct.**

You are currently at object\_dining\_table\_mella\_platform\_0.

And you are holding nothing.

Your available action space:

```
- ['go_to_object_hemnes_day_bed_platform_5', 'go_to_object_bin_6_platform_0',  
'go_to_object_Side_Table_317_1_platform_1',  
'go_to_object_side_table_strind_platform_0',  
'go_to_object_side_table_strind_platform_1',  
'go_to_object_Side_Table_203_1_platform_0',  
'go_to_object_Side_Table_203_1_platform_1', 'go_to_object_chair_antnas_platform_0',  
'go_to_object_dresser_birkeland_platform_0',  
'go_to_object_dresser_birkeland_platform_1',  
'go_to_object_dresser_birkeland_platform_2',  
'go_to_object_dresser_birkeland_platform_3',  
'go_to_object_dresser_birkeland_platform_4',  
'go_to_object_dresser_birkeland_platform_5',  
'go_to_object_dresser_birkeland_platform_6', 'go_to_object_Basketball_1_platform_0',  
'go_to_object_dining_table_mella_platform_0',  
'go_to_object_Floor_Lamp_13_platform_0',  
'go_to_object_armchair_emmabo_platform_0', 'go_to_object_BaseballBat_2_platform_0'] -  
['pick_object_1_of_current_platform', 'pick_object_2_of_current_platform',  
'pick_object_3_of_current_platform', 'pick_object_4_of_current_platform',  
'pick_object_5_of_current_platform', 'pick_object_6_of_current_platform'] -  
['show_receptacles_of_object_1_of_current_platform',  
'show_receptacles_of_object_2_of_current_platform',  
'show_receptacles_of_object_3_of_current_platform',  
'show_receptacles_of_object_4_of_current_platform',  
'show_receptacles_of_object_5_of_current_platform',  
'show_receptacles_of_object_6_of_current_platform']  
- 'CALL_END': Call end (terminate the task). -
```

'rotate\_observation\_view\_of\_current\_platform': Change the viewing angle to observe the scene from another perspective.

Choose one of the available actions.

All your current available actions are listed as follows (all are without quotes and square brackets). Please don't mix up it with the action space of previous steps.

Current task:

Move Remote to Bowl's rear-left receptacles

Steps used: 5/20. You can only take at most 20 steps, so hurry if you've almost used all of them!

Here are some useful suggestions for you to better finish the task:

1. Your extra actions do not affect the success of the task, you only need to keep your steps within the total step limit.
2. When you want to place an object but cannot find enough space, you can try changing your standing position to find the correct orientation, try combining multiple receptacles, or remove objects that are blocking your way first(although you generally do not need to do this). Specifically, for the task asking you put object to empty platforms, try combining adjacent receptacles may be very useful.
3. For tasks that involve placing objects near other objects, please remember to get a focused image of the target object and examine it carefully. This is important because the regions may not be intuitive.
4. Important: Don't output anything other than the action you want to take. You will only receive the same hint informing you your invalid action. If you make mistakes, just start output correct actions from the next prompt without any apologize.

[\* See Fig. 29]

Here are 2 images, named ['Idle\_5\_1\_out\_of\_2.png', 'Idle\_5\_2\_out\_of\_2.png'] respectively.

Their description is as follows: This image shows the platform you are currently on in one view, with regions or objects marked with numbers for you to place objects accurately. The platform is object\_dining\_table\_mella\_platform\_0.

If the platform has no objects, a 3x3 grid will be marked on the platform to help you place objects, with each grid marked in numbers from 1 to 9. If there are objects on the platform, guiding lines will indicate the object numbers. Note that if you see multiple images, it means the platform is too long to be captured in one image, and it's guaranteed that the same object will be marked in THE SAME COLOR GUIDELINES AND THE SAME NUMBER. If you think you can't determine the number of an item in one picture, you can go through other pictures and see if this item is marked more clearly.

-----  
VLM Response: CALL\_END

-----  
(call\_ended)

Task\_result:

Target object placed in wrong direction, expected: rear-left, found: rear-right.

Task\_status:

Failed

679

## References

- [1] Andrew Szot, Bogdan Mazouze, Omar Attia, Aleksei Timofeev, Harsh Agrawal, Devon Hjelm, Zhe Gan, Zsolt Kira, and Alexander Toshev. From multimodal llms to generalist embodied agents: Methods and lessons. *arXiv preprint arXiv:2412.08442*, 2024.
- [2] 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. *arXiv preprint arXiv:2311.12871*, 2023.
- [3] Dhruv Batra, Angel Chang, Sonia Chernova, Andrew Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, Manolis Savva, and Hao Su. Rearrangement: A Challenge for Embodied AI. *arXiv preprint*, 2020.
- [4] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.
- [5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.
- [6] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. *arXiv preprint*, 2023.
- [7] Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazouze, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. Large language models as generalizable policies for embodied tasks. In *The Twelfth International Conference on Learning Representations*, 2023.
- [8] Meta. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models, 2024. URL <https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/>. Accessed: 2025-02-15.
- [9] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.
- [10] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- [11] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12462–12469. IEEE, 2024.
- [12] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR, 2022.
- [13] Jiayuan Gu, Devendra Singh Chaplot, Hao Su, and Jitendra Malik. Multi-skill mobile manipulation for object rearrangement. *arXiv preprint arXiv:2209.02778*, 2022.
- [14] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [15] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al.  $\pi 0$ : A vision-language-action flow model for general robot control, 2024. URL <https://arxiv.org/abs/2410.24164>, 2024.

- [16] Zeyu Zhang, Sixu Yan, Muzhi Han, Zaijin Wang, Xinggang Wang, Song-Chun Zhu, and Hangxin Liu. M3bench: Benchmarking whole-body motion generation for mobile manipulation in 3d scenes. *arXiv preprint arXiv:2410.06678*, 2024.
- [17] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37: 100428–100534, 2025.
- [18] Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Yifan Xu, Xixuan Song, Shudan Zhang, Hanyu Lai, Xinyi Liu, Hanlin Zhao, et al. Visualagentbench: Towards large multimodal models as visual foundation agents. *arXiv preprint arXiv:2408.06327*, 2024.
- [19] Jae-Woo Choi, Youngwoo Yoon, Hyobin Ong, Jaehong Kim, and Minsu Jang. Lotabench: Benchmarking language-oriented task planners for embodied agents. *arXiv preprint arXiv:2402.08178*, 2024.
- [20] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025.
- [21] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023.
- [22] Zhili Cheng, Yuge Tu, Ran Li, Shiqi Dai, Jinyi Hu, Shengding Hu, Jiahao Li, Yang Shi, Tianyu Yu, Weize Chen, et al. Embodiedeval: Evaluate multimodal llms as embodied agents. *arXiv preprint arXiv:2501.11858*, 2025.
- [23] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [24] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.
- [25] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.
- [26] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.
- [27] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.
- [28] Mengkang Hu, Pu Zhao, Can Xu, Qingfeng Sun, Jianguang Lou, Qingwei Lin, Ping Luo, Saravan Rajmohan, and Dongmei Zhang. Agentgen: Enhancing planning abilities for large language model based agent via environment and task generation. *arXiv preprint arXiv:2408.00764*, 2024.
- [29] Emily Jin, Jiaheng Hu, Zhuoyi Huang, Ruohan Zhang, Jiajun Wu, Li Fei-Fei, and Roberto Martín-Martín. Mini-behavior: A procedurally generated benchmark for long-horizon decision-making in embodied ai. *arXiv preprint arXiv:2310.01824*, 2023.



- [30] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056. PMLR, 2020.
- [31] Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. *arXiv preprint arXiv:2002.12292*, 2020.
- [32] Kuan Fang, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Adaptive procedural task generation for hard-exploration problems. *arXiv preprint arXiv:2007.00350*, 2020.
- [33] Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models. *arXiv preprint arXiv:2310.01361*, 2023.
- [34] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Motlaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.
- [35] Abby O’Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.
- [36] Andrew Szot, Karmesh Yadav, Alex Clegg, Vincent-Pierre Berges, Aaron Gokaslan, Angel Chang, Manolis Savva, Zsolt Kira, and Dhruv Batra. Habitat rearrangement challenge 2022. [https://aihabitat.org/challenge/2022\\_rearrange](https://aihabitat.org/challenge/2022_rearrange), 2022.
- [37] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021.
- [38] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents, 2025. URL <https://arxiv.org/abs/2502.09560>.
- [39] Ahmed Jaafar, Shreyas Sundara Raman, Yichen Wei, Sudarshan Harithas, Sofia Juliani, Anneke Wernerfelt, Benedict Quartey, Ifrah Idrees, Jason Xinyu Liu, and Stefanie Tellex.  $\lambda$ : A benchmark for data-efficiency in long-horizon indoor mobile manipulation robotics, 2025. URL <https://arxiv.org/abs/2412.05313>.
- [40] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, and Shyamal Anadkat et al. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.
- [41] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, and et al. Anja Hauth. Gemini: A family of highly capable multimodal models, 2024. URL <https://arxiv.org/abs/2312.11805>.
- [42] Anthropic. About claude models. <https://docs.anthropic.com/en/docs/about-claude/models>, 2024. Accessed: 2024-09-03.
- [43] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020.
- [44] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

- [45] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- [46] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR, 2023.
- [47] Simon Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Peter Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37: 110935–110971, 2024.
- [48] Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290, 2024.
- [49] Jiayi Pan, Yichi Zhang, Nicholas Tomlin, Yifei Zhou, Sergey Levine, and Alane Suhr. Autonomous evaluation and refinement of digital agents. *arXiv preprint arXiv:2404.06474*, 2024.
- [50] Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*, 3, 2024.
- [51] Ram Ramrakhyia, Matthew Chang, Xavier Puig, Ruta Desai, Zsolt Kira, and Roozbeh Mottaghi. Grounding multimodal llms to embodied agents that ask for help with reinforcement learning. *arXiv preprint arXiv:2504.00907*, 2025.
- [52] Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*, 2024.
- [53] Andrew Szot, Bogdan Mazouze, Harsh Agrawal, R Devon Hjelm, Zsolt Kira, and Alexander Toshev. Grounding multimodal large language models in actions. *Advances in Neural Information Processing Systems*, 37:20198–20224, 2024.
- [54] Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization for llm agents. *arXiv preprint arXiv:2403.02502*, 2024.
- [55] Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*, 2023.
- [56] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.
- [57] Jake Grigsby, Yuke Zhu, Michael Ryoo, and Juan Carlos Niebles. Vlm q-learning: Aligning vision-language models for interactive decision-making. *arXiv preprint arXiv:2505.03181*, 2025.
- [58] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [59] Thomas Carta, Clément Romic, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. Grounding large language models in interactive environments with online reinforcement learning. In *International Conference on Machine Learning*, pages 3676–3713. PMLR, 2023.
- [60] Amazon. Amazon mechanical turk. <https://www.mturk.com/>, 2024.

- 874 [61] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias  
875 Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the*  
876 *IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.