

Supplementary Material

Guardian: Detecting Robotic Planning and Execution Errors with Vision-Language Models

Paul Pacaud, Ricardo Garcia Pinel, Shizhe Chen, Cordelia Schmid
Inria, École normale supérieure, CNRS, PSL Research University
firstname.lastname@inria.fr

A Real-Robot Experiments

We conduct real-robot experiments using Guardian to monitor the planning and execution performance of the 3DLotus++ [1] policy during tabletop manipulation tasks. First, to ensure reproducibility and robust evaluation across diverse task variations, we collect a real-world dataset for offline analysis of execution and planning failure detection in Section A.1. Subsequently, we demonstrate the real-time effectiveness of Guardian through online experiments on three manipulation tasks in Section A.2. Experiments are performed using a 6-DoF UR5 robotic arm equipped with three RealSense D435 cameras as shown in Fig. 1.

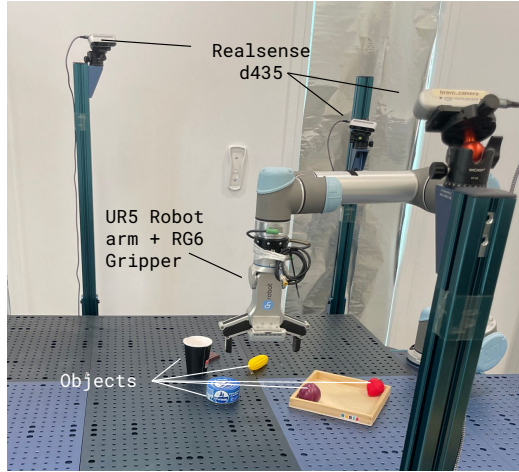


Figure 1: Our real-robot setup includes a UR5 robotic arm equipped with a RG6 gripper. Three RealSense D435 cameras are mounted around the table.

A.1 Offline Real-Robot Experiments

Split	Task Variations	UR5-Fail	
		Execution Samples	Planning Samples
Train	7	404	224
Val	6	30	10
Test	22	100	74

Table 1: UR5-Fail dataset distribution across train, validation, and test splits.

Dataset Construction. We construct an offline dataset, named **UR5-Fail**, by rolling out the 3DLo-tus++ policy [1] across 32 distinct task variations, recording initial and final images for each subtask. Splits are mutually exclusive except for three overlapping task variations present in both training and validation sets. To augment this dataset, we include additional demonstrations obtained via teleoperation. Each subtask is manually annotated as a success (ground truth) or categorized into specific failure modes: (1) **no_close**, where the gripper fails to close during grasping; (2) **translation**, involving unintended positional offsets during grasping or object placement; and (3) **wrong_object**, where an incorrect object is manipulated. For planning evaluations, we annotate ground truth plans and introduce perturbations using our failure generation pipeline described previously. The resulting dataset, UR5-Fail (Table 1), contains carefully curated examples with balanced training, validation, and test splits. This dataset enables the evaluation and fine-tuning of Guardian within our real-robot setup.

We will open-source this real-robot dataset.

Model	UR5-Fail Test	
	Execution	Planning
GPT-4.1	0.75	0.89
GPT-4.1-mini	0.56	0.77
InternVL2.5-8B	0.53	0.84
Guardian	0.74	0.92
Guardian-RealRobot	0.85	0.93

Table 2: Performance comparison on the UR5-Fail test set.

Evaluation. Table 2 compares the performance of Guardian, Guardian fine-tuned on the real robot (Guardian-RealRobot), its base model InternVL2.5-8B, and GPT-4.1 on UR5-Fail test set. Guardian effectively generalizes to this dataset, significantly outperforming its base model. While Guardian achieves similar performance to GPT-4.1 on execution tasks, it notably surpasses GPT-4.1 on planning tasks despite being smaller. Furthermore, fine-tuning Guardian for our real-robot setup (Guardian-RealRobot) provides substantial performance improvements in both execution and planning.

A.2 Online Real-Robot Experiments

Table 3: Success rate for online task evaluation with Guardian integration. Results show improved robustness across three manipulation tasks, particularly in perturbed conditions.

Guardian	Put food in box		Put fruits in plates		Stack cups	
	Normal runs	Perturbed runs	Normal runs	Perturbed runs	Normal runs	Perturbed runs
×	0.60	0.00	0.60	0.20	0.80	0.00
✓	0.80	0.80	0.60	0.40	1.00	0.40

To validate Guardian’s utility in live scenarios, we integrate Guardian-RealRobot into the 3DLo-tus++ [1] robotic pipeline and evaluate its impact on three unseen manipulation tasks: putting food (tuna and corn) into a box, putting fruits (grapes and banana) on plates, and stacking cups (pink onto yellow). We assess the success rate (%) of the 3DLo-tus++ [1] policy both with and without Guardian across ten episodes per task—five under normal conditions and five subjected to human-induced perturbations. These perturbations are designed to evaluate the robotic policy’s ability to replan and recover effectively from erroneous previous actions, rather than blindly continuing its initial plan. Each episode involves exactly one of the following perturbations:

- **Translation:** During a ”grasp” action, a human slightly shifts the object being manipulated, causing the gripper to fail to grasp the intended object. Similarly, during a ”move to” action,

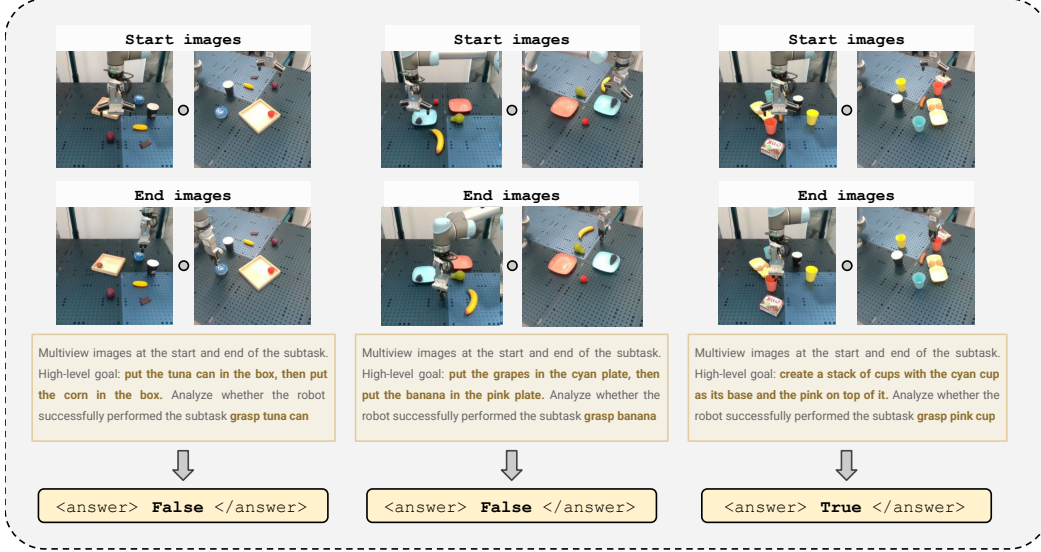


Figure 2: Online Real-Robot Experiments.

We show one example for each of the three tasks used for online experiments. Each example consists in six images and a query (three viewpoint images for the start of the subtask and three viewpoint images for the end).

the target location is slightly altered, resulting in the gripper moving to an incorrect or empty location.

- **Object Swaps:** During a "grasp" action, the intended object is swapped by a human for another object, causing the gripper to grasp the incorrect object. Likewise, during a "move to" action, the target object or location is exchanged with another object or location, leading the gripper to move towards an unintended destination.

Results are detailed in Table 3, with examples of execution scenarios presented in Fig. 2 for the three tasks evaluated. We also provide a video showcasing each task.

During task execution, the 3DLotus++ [1] policy generates an initial plan and executes subtasks sequentially. Although the motion planner can dynamically adapt trajectories within a subtask, it does not verify subtask completion before proceeding. Guardian addresses this by verifying the success of each subtask completion and triggering a replanning mechanism when necessary, returning to the initial gripper position of the subtask to retry. Consequently, Guardian yields moderate improvements in success rates under normal conditions (+20%, +0%, and +20%) and substantial enhancements under perturbations (+80%, +20%, and +40%).

A.3 Failure Cases

Despite these promising results, certain limitations remain. Guardian occasionally introduces false-negative detections, erroneously indicating failure and causing unintended object releases or incorrect positioning, negating some benefits (see the false negative detection on the left of Fig. 3). Furthermore, even in cases where the failure detection is correct, the state of the scene may not be recoverable by simply retrying the current subtask, leading to no performance gain in these scenarios. For instance, on the right of Fig. 3, Guardian detects a failure at the current subtask "move grasped object to cyan cup" because the pink cup fell on the table. Simply retrying "move grasped object to cyan cup" would not work here as the robot first needs to restart the previous subtask "grasp pink cup". Addressing these replanning challenges would further enhance Guardian's efficacy in complex, real-world robotic tasks.

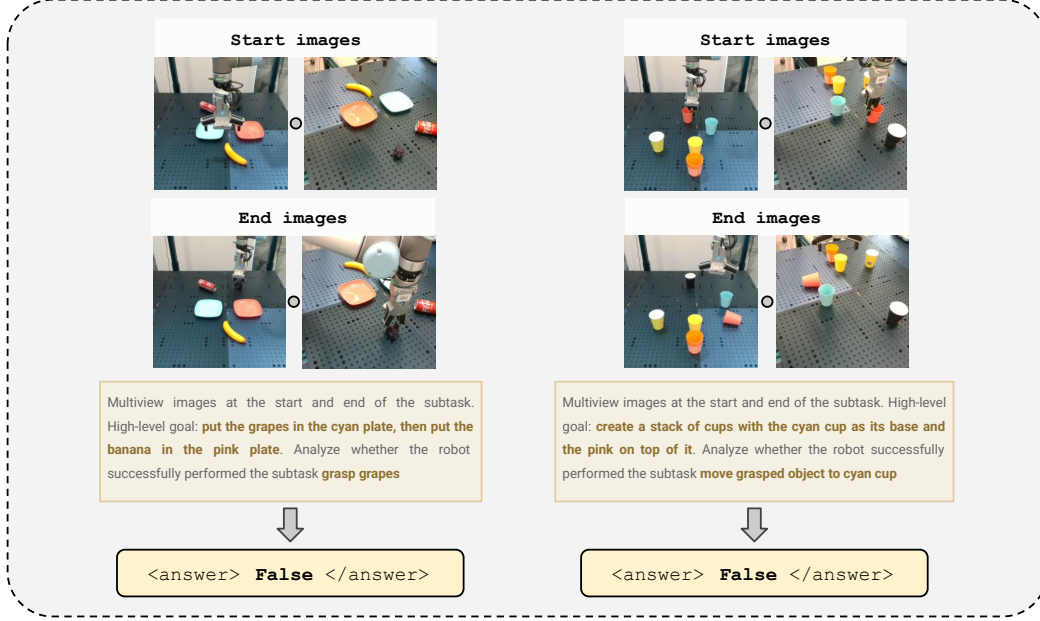


Figure 3: Failure Cases of Guardian during Online Real-Robot Experiments.

B Additional Details on the Datasets

B.1 Examples of execution and planning samples from RLBench-Fail and BridgeDataV2-Fail

From RLBench-Fail and BridgeDataV2-Fail, we show execution samples in Fig. 4 and planning samples in Fig. 5. Each sample is a pair of (optionally multiview) images at the start and end of a subtask, along with a textual prompt describing the current situation.

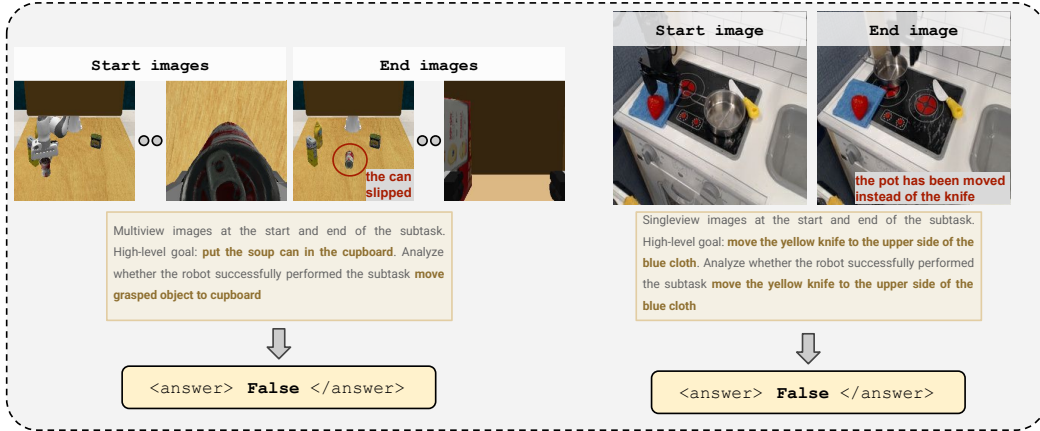


Figure 4: (Left) Execution Failure from RLBench-Fail (Right) Execution Failure from BridgeDataV2-Fail

B.2 Failure Modes Distribution

In Fig. 6, we present the distribution of failure modes automatically generated by our pipeline across RLBench-Fail and BridgeDataV2-Fail datasets. The execution and planning datasets contain a balanced proportion of positive samples (ground truth) and negative samples (perturbations). Some failure modes occur more frequently than others; for instance, translation perturbations appear more

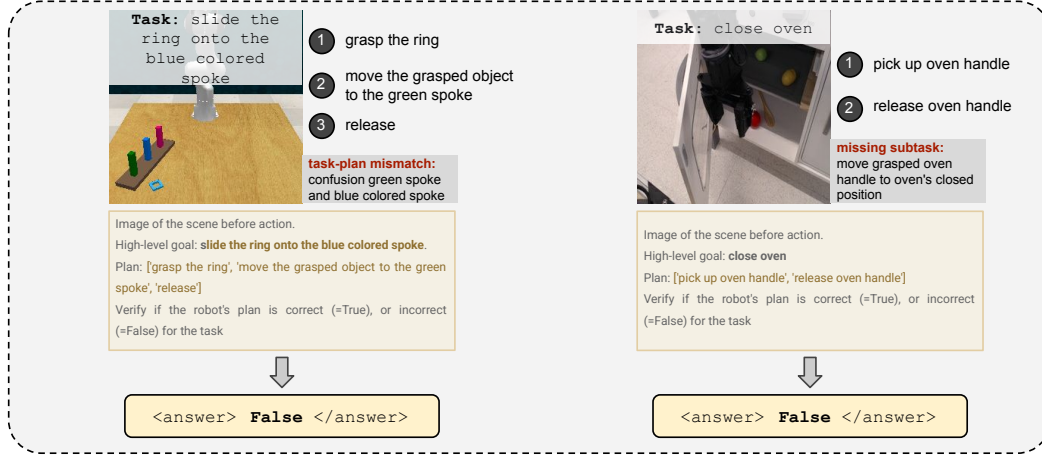


Figure 5: (Left) Planning Failure from RL Bench-Fail (Right) Planning Failure from BridgeDataV2-Fail

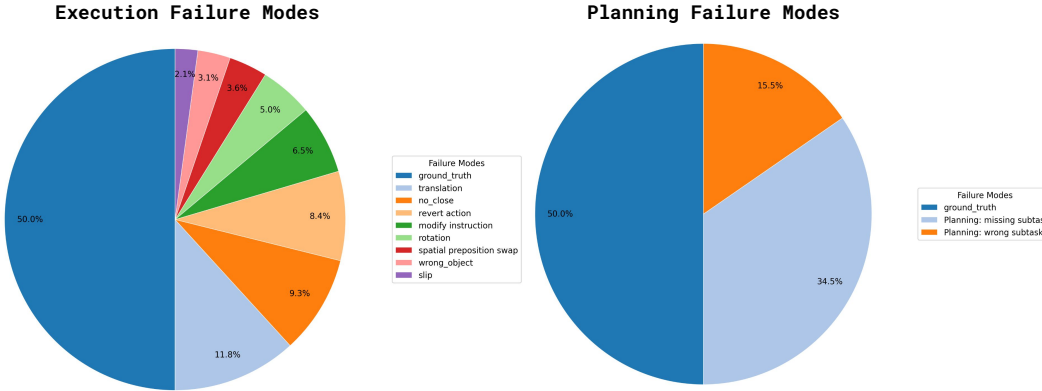


Figure 6: Failure Modes Distribution.

Distribution of the failure modes for planning and execution samples across RL Bench-Fail and BridgeDataV2-Fail.

often than slips. This discrepancy arises because translation perturbations can be universally applied across all tasks, whereas slip perturbations are restricted to tasks involving the movement of grasped objects.

B.3 BridgeDataV2 - Pre-Processing of ECoT annotations

Here, we briefly describe the automated cleaning procedure applied to the ECoT annotations. The rationale behind utilizing ECoT annotations lies in their ability to provide detailed, frame-by-frame annotations for subtasks, including visible scene objects. However, as these annotations are automatically generated via pipelines utilizing early-generation LLMs and VLMs such as Gemini 1.0, they are susceptible to errors. Such inaccuracies can impact our automatic failure-generation pipeline. To mitigate these issues, we employ a combination of heuristics and an advanced LLM (Mistral-Small-24B). First, the LLM identifies the ground truth item manipulated within task instructions (e.g., extracting "apple" from "put the apple in the fridge"). Using this identified ground truth object, we discard ECoT episodes where this object does not appear among the visible objects listed in the ECoT annotations. Additionally, we further refine the dataset by removing episodes with bounding boxes that are either excessively small or positioned too close to the image edges, as these typically indicate incorrect object grounding or ambiguity that we prefer to exclude from our dataset.

B.4 RLBench-Fail Task Variations

We provide the full list of unique task variations of each split of RLBench-Fail. Note that some variation names are very similar (e.g, close_fridge+0 and close_fridge2+0) but they contain different objects (e.g, different kind of fridges).

Train Execution Task Variations with Instructions

Task Variation	Task Instructions
close_fridge+0	swing the fridge1 door shut
close_fridge2+0	swing the fridge2 door shut
close_jar_peract+15	grasping the lid, lift it from the table and use it to seal the azure jar
close_jar_peract+16	grasping the lid, lift it from the table and use it to seal the violet jar
close_laptop_lid+0	shut the laptop lid1
close_laptop_lid2+0	shut the laptop lid2
close_microwave+0	push the microwave1 door shut
insert_onto_square_peg_peract+0	slide the ring onto the red colored spoke
insert_onto_square_peg_peract+1	slide the ring onto the maroon colored spoke
insert_onto_square_peg_peract+2	slide the ring onto the lime colored spoke
light_bulb_in_peract+17	pick up the light bulb from the rose stand, lift it up to just above the lamp, then screw it down into the lamp in a clockwise fashion
light_bulb_in_peract+19	pick up the light bulb from the white stand, lift it up to just above the lamp, then screw it down into the lamp in a clockwise fashion
meat_off_grill_peract+0	remove the chicken from the grill and set it down to the side
open_box+0	turn the attached lid to open the box1
open_box2+0	turn the attached lid to open the box2
open_door+0	grasp the handle firmly, twist it, and push to open the door1
open_door2+0	grasp the handle firmly, twist it, and push to open the door2
open_drawer+0	grip the bottom handle and pull the bottom drawer1 open
open_drawer+1	grip the middle handle and pull the middle drawer1 open
open_drawer+2	grip the top handle and pull the top drawer1 open
open_drawer_long+0	grip the bottom handle and pull the bottom drawer1 open
open_drawer_long+1	grip the middle bottom handle and pull the middle bottom drawer1 open
open_microwave+0	use the microwave1 door handle to swing the microwave1 door open
pick_and_lift+0	pick up the red block and lift it up to the target
pick_and_lift+14	pick up the teal block and lift it up to the target
pick_and_lift+16	pick up the violet block and lift it up to the target
pick_and_lift+2	pick up the lime block and lift it up to the target
pick_and_lift+7	pick up the cyan block and lift it up to the target
pick_and_lift_cylinder+0	pick up the red cylinder and lift it up to the target
pick_up_cup+10	grasp the gray cup and lift it
pick_up_cup+11	grasp the orange cup and lift it
pick_up_cup+8	grasp the magenta cup and lift it
pick_up_cup+9	grasp the silver cup and lift it
place_cups_peract+2	pick up 3 mugs and slide their handles onto the cup holder spokes

Task Variation	Task Instructions
place_shape_in_shape_sorter_peract+0	place the cube into its slot in the shape sorter
place_shape_in_shape_sorter_peract+1	place the cylinder into its slot in the shape sorter
place_shape_in_shape_sorter_peract+2	place the triangular prism into its slot in the shape sorter
place_shape_in_shape_sorter_peract+3	place the star into its slot in the shape sorter
place_wine_at_rack_location_peract+0	slide the bottle onto the middle part of the rack
place_wine_at_rack_location_peract+1	slide the bottle onto the left part of the rack
push_button+0	press the button with the maroon base
push_button+13	press the button with the azure base
push_button+3	press the button with the navy base
push_button+4	press the button with the yellow base
push_buttons4+1	push down the button with the navy base, then the teal one
push_buttons4+2	press the green button, then press the yellow button, then press the rose button
put_groceries_in_cupboard+0	pick up the crackers box and place it in the cupboard
put_groceries_in_cupboard+3	pick up the soup can and place it in the cupboard
put_groceries_in_cupboard+7	pick up the mustard bottle and place it in the cupboard
put_item_in_drawer_peract+0	open the bottom drawer1 and place the cube inside of it
put_item_in_drawer_peract+1	open the middle drawer1 and place the cube inside of it
put_items_in_drawer+0	open the bottom drawer1 and place the block inside of it, then place the cylinder inside and finally the moon
put_items_in_drawer+2	open the top drawer1 and place the block inside of it, then place the cylinder inside and finally the moon
put_money_in_safe+0	place the stack of bank notes on the bottom shelf of the safe
put_money_in_safe+1	place the stack of bank notes on the middle shelf of the safe
reach_and_drag_peract+14	grasping the stick by one end, pick it up and use the its other end to move the block onto the teal target
reach_and_drag_peract+18	grasping the stick by one end, pick it up and use the its other end to move the block onto the black target
reach_and_drag_peract+5	grasping the stick by one end, pick it up and use the its other end to move the block onto the navy target
slide_block_to_color_target_peract+0	cover the green target with the block by pushing the block in its direction
slide_block_to_color_target_peract+1	cover the blue target with the block by pushing the block in its direction
slide_block_to_color_target_peract+2	cover the pink target with the block by pushing the block in its direction
stack_blocks+24	arrange 2 magenta blocks in a vertical stack on the table top
stack_blocks+30	arrange 2 gray blocks in a vertical stack on the table top
stack_blocks+36	arrange 2 olive blocks in a vertical stack on the table top
stack_blocks+39	arrange 2 purple blocks in a vertical stack on the table top
stack_cups+0	keeping the red cup on the table, stack the other two onto it
stack_cups_peract+1	keeping the maroon cup on the table, stack the other two onto it
stack_cups_peract+2	keeping the lime cup on the table, stack the other two onto it
sweep_to_dustpan_of_size_peract+0	use the broom to brush the dirt into the tall dustpan
turn_tap_peract+0	grasp the left tap and turn it

Table 4: RLBench-Fail train set

Val Execution Task Variations with Instructions

Task Variation	Task Instructions
close_jar_peract+3	grasping the lid, lift it from the table and use it to seal the green jar
close_microwave2+0	push the microwave2 door shut
light_bulb_in_peract+1	pick up the light bulb from the maroon stand, lift it up to just above the lamp, then screw it down into the lamp in a clockwise fashion
open_drawer2+0	grip the bottom handle and pull the bottom drawer2 open
open_drawer_long+2	grip the middle top handle and pull the middle top drawer1 open
pick_and_lift+18	pick up the black block and lift it up to the target
pick_and_lift_star+0	pick up the red star and lift it up to the target
pick_up_cup+12	grasp the olive cup and lift it
place_cups_peract+0	pick up one cup and slide its handle onto a spoke on the mug holder
push_button+15	press the button with the rose base
put_money_in_safe+2	place the stack of bank notes on the top shelf of the safe
stack_blocks+27	arrange 2 silver blocks in a vertical stack on the table top

Table 5: RLBench-Fail val set

Test Execution Task Variations with Instructions

Task Variation	Task Instructions
close_box+0	rotate the attached lid until the box1 is closed and sealed
close_door+0	grasp the handle and pull the door towards you until it fully closes
close_drawer+0	press on the bottom drawer1 until it is closed
close_grill+0	lower the grill cover using the handle to close the bbq
close_jar_peract+4	grasping the lid, lift it from the table and use it to seal the blue jar
insert_onto_square_peg_peract+3	slide the ring onto the green colored spoke
lamp_on+0	close the gripper and press on the button until the light turns on
light_bulb_in_peract+2	pick up the light bulb from the lime stand, lift it up to just above the lamp, then screw it down into the lamp in a clockwise fashion
meat_off_grill_peract+1	remove the steak from the grill and set it down to the side
open_drawer3+0	grip the bottom handle and pull the bottom drawer3 open
open_drawer_long+3	grip the top handle and pull the top drawer1 open
open_fridge+0	grip the handle and slide the fridge1 door open
pick_and_lift_moon+0	pick up the red moon and lift it up to the target
pick_and_lift_toy+0	pick up the red rubber duck and lift it up to the target
pick_up_cup+13	grasp the purple cup and lift it
place_cups_peract+1	pick up 2 mugs and slide their handles onto the cup holder spokes
place_shape_in_shape_sorter_peract+4	place the moon into its slot in the shape sorter
place_wine_at_rack_location_peract+2	slide the bottle onto the right part of the rack
push_button+17	press the button with the white base
push_buttons4+3	press the maroon button, then press the blue button, then press the orange button, then press the magenta button

Task Variation	Task Instructions
put_all_groceries_in_cupboard+0	put the crackers box, the chocolate jello box, strawberry jello box, soup can, spam can, mustard bottle and sugar box in the cupboard
put_cube_in_safe+0	put the cube away in the safe on the bottom shelf
put_groceries_in_cupboard+8	pick up the sugar box and place it in the cupboard
put_item_in_drawer_peract+2	open the top drawer1 and place the cube inside of it
put_items_in_drawer+4	open the middle drawer1 and place the block inside of it, then place the moon inside and finally the cylinder
reach_and_drag_peract+7	grasping the stick by one end, pick it up and use the its other end to move the block onto the cyan target
slide_block_to_color_target_peract+3	cover the yellow target with the block by pushing the block in its direction
stack_blocks+33	arrange 2 orange blocks in a vertical stack on the table top
stack_cups+3	keeping the green cup on the table, stack the other two onto it
sweep_to_dustpan_of_size_peract+1	use the broom to brush the dirt into the short dustpan
take_shoes_out_of_box+0	grasp the edge of the box1 lid to open it, then grasp each shoe, lifting up out of the shoe box and leaving them down on the table
toilet_seat_up+0	grip the edge of the toilet seat and lift it up to an upright position
tower4+1	pick the orange cube and put it on the green marker, then stack the gray block on top of the orange, then stack the lime block on top of the gray, finally stack the rose block on top of the stacked cubes
tower4+3	pick the white cube and put it on the green marker, then stack the teal block on top of the white, finally stack the blue block on top of the stacked cubes
turn_tap_peract+1	grasp the right tap and turn it

Table 6: RL Bench-Fail test set

References

- [1] R. Garcia, S. Chen, and C. Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and LLM-guided 3D policy. In *ICRA*, 2025.