THE COLOSSEUM APPENDIX

I. PERTURBATION FACTORS SELECTION RATIONALE

We study recent real-world diverse robot datasets, such as, Open-X [**?**], DROID [**?**], Ego4D [**?**], and conclude that our identified factors indeed exist in these datasets. In Figure 1, randomly sampled from DROID dataset, we can observe that `MO_Color/Texture/Size`, `Light_Color`, `Table_Color/Texture`, `Distractors`, `Camera_Pose`, `Background` changing across scenes. While it is not explicity reported, we can also infer that mass of cups would also change. While these factors do not cover the exhaustive list of factors that vary in the real-world, our empirical analysis shows that THE COLOSSEUM factors do affect the SoTA robot manipulation models, and hence are important to study. It is challenging to breakdown real-world into an exhaustive systematic enumeration of factors. THE COLOSSEUM is one of the first attempt towards increasing real-world task robustness for robotic manipulation via such a systematic purturbation benchmark.

II. SIMULATION TASK DETAILS

We describe each of the 20 tasks in detail, along with their RLBench variations and success condition.

A. open drawer

Filename: `open_drawer.py`

Task: Open one of the three drawers: `bottom`, `middle`, or `top`. **Success Metric**: The prismatic joint of the specified drawer is fully extended.

B. slide block to target

Filename: `slide_block_to_target.py`

Task: Slide the block to square target. **Success Metric**: Some part of the block is inside the specified target area.

C. basketball in hoop

Filename: `basketball_in_hoop.py`

Task: Pick up the basketball and put it into the hoop. **Success Metric**: 1 basketball falls into the hoop.

D. meat on grill

Filename: `meat_on_grill.py`

Task: Take either the `chicken` or `steak` off the rack and put it on the grill. **Success Metric**: The specified meat is on the grill.

E. close box

Filename: `close_box.py`

Task: Close the box. **Success Metric**: The revolute joint of the specified handle is at least 60° off from the starting position.

F. close laptop lid

Filename: `Close_Laptop_Lid.py`

Task: Close the laptop lid. **Success Metric**: The revolute joint of the specified handle is at least 60° off from the starting position.

G. empty dishwasher

Filename: `empty_dishwaser.py`

Task: Open the dishwasher and take out the plate. **Success Metric**: The plate has been taken out of the dishwasher.

H. reach and drag

Filename: `reach_and_drag.py`

Task: Grab the stick and use it to drag the cube on to the target square. **Success Metric**: Some part of the block is inside the specified target area.

I. get ice from fridge

Filename: `get_ice_from_fridge.py`

Task: Pick up the cup and push it against the ice dispenser. **Success Metric**: Cup pushed against the ice dispenser.

J. hockey

Filename: `hockey.py`

Task: Pick up the hockey stick, and hit the ball into the goal pose. **Success Metric**: The ball enters into the goal pose.

K. put money in safe

Filename: `put_money_in_safe.py`

Task: Pick up the stack of money and put it inside the safe on the specified shelf. The shelf has three placement locations: `top`, `middle`, `bottom`. **Success Metric**: The stack of money is on the specified shelf inside the safe.

L. place wine at rack location

Filename: `place_wine_at_rack_location.py`

Task: Grab the wine bottle and put it on the wooden rack at one of the three specified locations: `left`, `middle`, `right`. The locations are defined with respect to the orientation of the wooden rack. **Success Metric**: The wine bottle is at the specified placement location on the wooden rack.

M. move hanger

Filename: `move_hanger.py`

Task: Pick up the hanger and move it from one side to another. **Success Metric**: The hanger is successfully hooked onto the other hanger holder.

N. wipe desk

Filename: `wipe_desk.py`

Task: Pick up the sponge and wipe the dust particles off the desk. **Success Metric**: The table is being cleaned up.

O. straighten rope

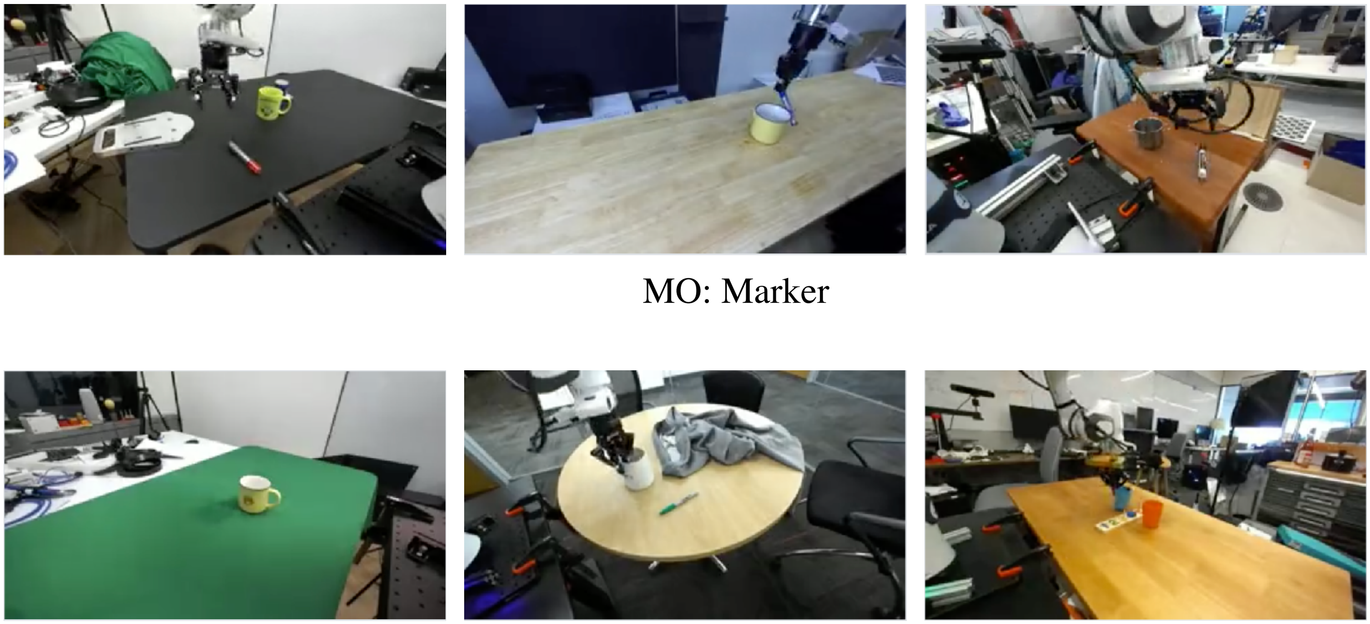
Filename: `straighten_rope.py`

Task: Pick up one end of the rope and move it to the nearest tape patch, and the same for the other end. **Success Metric**: The two patches have one side of the rope on each.

P. insert onto square peg

Filename: `insert_onto_square_peg.py`

Task: Pick up the square and put it on the specified color spoke. The spoke colors are sampled from the full set of 20 color instances. **Success Metric**: The square is on the specified spoke.



MO: Marker

MO: Cup

Fig. 1: Dataset samples from DROID showing scene variations including MO_Color/Texture/Size, Light_Color, Table_Color/Texture, Distractors, Camera_Pose, Background, supporting our choice perturbation_factors.

Q. stack cups

Filename: stack_cups.py

Task: Stack all cups on top of the specified color cup. The cup colors are sampled from the full set of 20 color instances. The scene always contains three cups. **Success Metric**: All other cups are inside the specified cup.

R. turn oven on

Filename: turn_oven_on.py

Task: Grasp onto the knob and turn it on. **Success Metric**: The knob is turned on.

S. setup chess

Filename: setup_chess.py

Task: Pick up the odd chess pieces and put it into the start position. **Success Metric**: The odd one out chess piece has been placed on the designated spot.

T. scoop with spatula

Filename: Scoop_with_Spatula.py

Task: Pick up the spatula and scoop up the cube. **Success Metric**: The cube has been successfully picked up using the spatula.

III. Simulation details

We provide full benchmark perturbation details for each task in Tables II. Tables II defines MO and RO objects for each task, specifies whether the applied perturbation is sampled from a discrete set or a continuous range, and finally provides the corresponding set size or the range. '-' means the perturbation does not apply due to either absence of RO for the task, or the simulator doesn't support that factor for the specified object. The remaining 6 perturbation_factors apply to all the tasks. We specify their corresponding perturbation parameters in the main text (Section III.C). In Figure 2, we show an example of a task configuration file, and how its perturbation_factors and their parameters can be specified or changed. In Figures 3- 22, we show all perturbed views for each task.

A. Training details and Detailed results

To train the baseline models, we use 1-4 NVIDIA RTX A6000 for 1-6 days. For a full evaluation over The Colosseum, we run multiple parallel jobs with batches launching in a sequence. Total compute used for this process was 4 NVIDIA RTX A6000 over 2-3 days for each model.

We report detailed per task success rates on each of the perturbation_factors in Tables III-VI for all the baselines.

IV. Real world details

A. Robot hardware setup

The real-robot experiments use a Franka Panda manipulator with a parallel gripper. For perception, we use a Kinect-2 RGB-D camera mounted on a tripod, at an angle, pointing towards the tabletop. Kinect-2 provides RGB-D images of resolution 512 × 424 at 30Hz. The extrinsic between the camera and robot base-frame are calibrated with the easy hand-eye package. We use an ARUCO AR marker mounted on the gripper to aid the calibration process, as shown in Figure 23.

B. Task setup

For the object assets, we 3D printed all of them as shown in Figure 24. For (RO/MO-Sizes), we vary the scale by ±0.2 times original object size. For RO/MO-Colors, we use two different printing filaments (red and blue). For the Light-Color variation, we use a single color-changing spotlight. The success condition of each of the tasks are defined as follows:

- `slide block to target`: Push the colored block into the light yellow patch with the word 'target' written on it.
- `setup chess`: Pick up the pawn piece and put it onto the blue marked chess spot.
- `insert on square peg`: Pick up the colored square peg and insert it onto the right most pole.
- `scoop with spatula`: Push the spatula inwards to scoop up the cube, and then lift it up.

C. Data collection

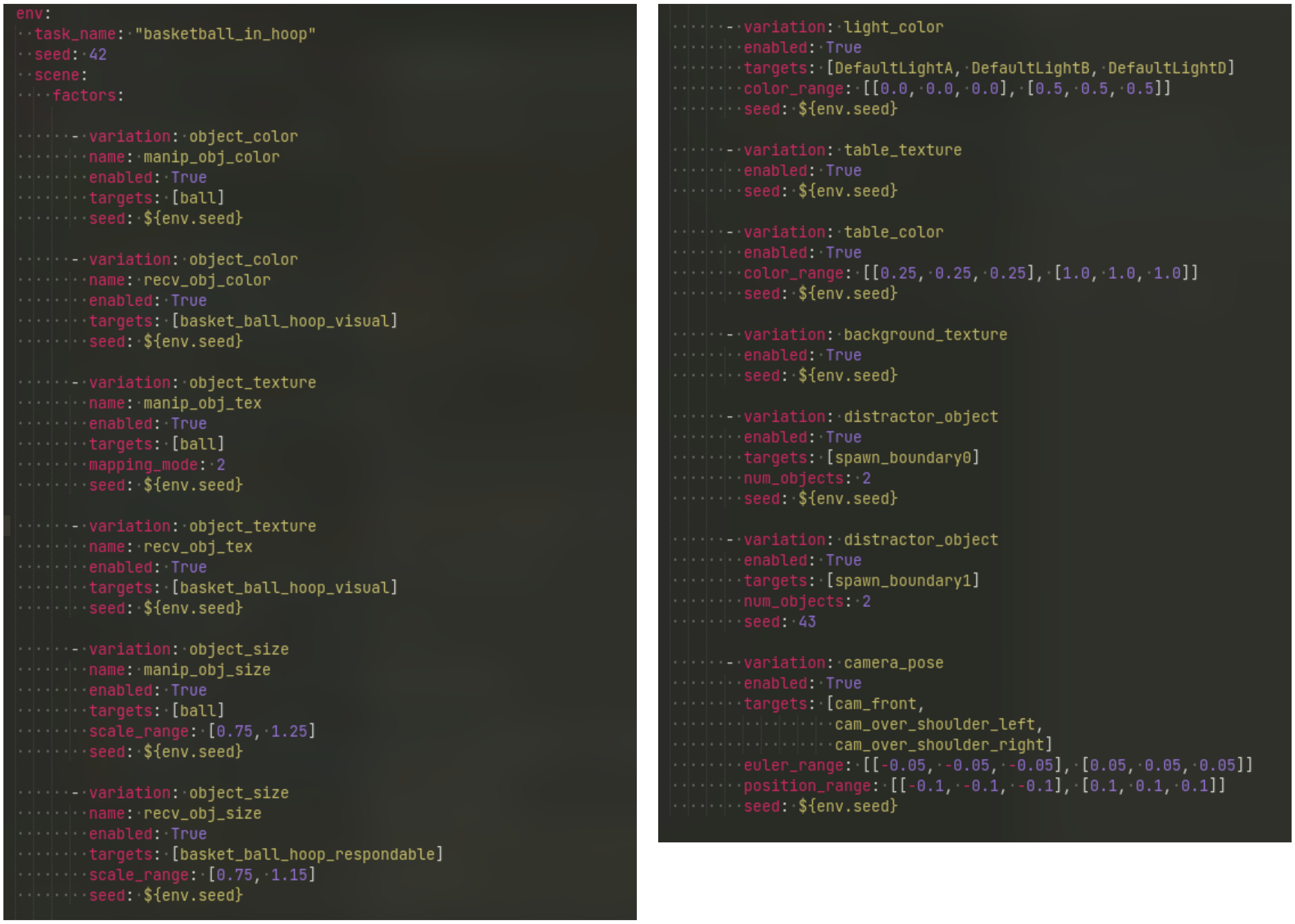
We gather data through demonstrations using an HTC Vive controller, a device capable of 6 degrees of freedom (DoF) tracking, ensuring precise positioning relative to a stationary base station. The positions captured are visualized in RViz as markers on the real-time RGB-D pointcloud data obtained from the Kinect camera sensor. Users determine desired positions to record as keypoints by referencing both the marker and the pointcloud. These specified positions are then realized through the employment of a motion planning algorithm. For this purpose, we employ the Franka ROS interface along with MoveIt, which inherently utilizes the RRT-Connect planning algorithm by default.

D. Training and Evaluation details

The real robot's training was run on 1 NVIDIA TITAN RTX GPU for 1 day. We monitor the keypoints predicted by the real-world model to verify the safety of the next action. The robot continues to execute predicted keypoints during evaluation unless manually halted by the human operator.

E. Ablation study

To investigate the compound effects of multiple perturbations on model performance and their correlation with real-world scenarios, we conducted an ablation study using the task `slide_block_to_target`. We selected three perturbation combinations from real-world experiments and constructed three analogous real-world scenarios: a workbench, a dining table, and a study room table. Each scenario was subjected to the same perturbations derived from the benchmark's combinations. We assessed `PerAct`, a model trained on these real-world experiments, across both sets of scenarios. Each scenario underwent 10 episodes across five trials. Our analysis revealed a strong correlation between two of the three scenarios, as detailed in the depicted in Table I.



```yaml
env:
  task_name: "basketball_in_hoop"
  seed: 42
  scene:
    factors:

      - variation: object_color
        name: manip_obj_color
        enabled: True
        targets: [ball]
        seed: ${env.seed}

      - variation: object_color
        name: recv_obj_color
        enabled: True
        targets: [basket_ball_hoop_visual]
        seed: ${env.seed}

      - variation: object_texture
        name: manip_obj_tex
        enabled: True
        targets: [ball]
        mapping_mode: 2
        seed: ${env.seed}

      - variation: object_texture
        name: recv_obj_tex
        enabled: True
        targets: [basket_ball_hoop_visual]
        seed: ${env.seed}

      - variation: object_size
        name: manip_obj_size
        enabled: True
        targets: [ball]
        scale_range: [0.75, 1.25]
        seed: ${env.seed}

      - variation: object_size
        name: recv_obj_size
        enabled: True
        targets: [basket_ball_hoop_respondable]
        scale_range: [0.75, 1.15]
        seed: ${env.seed}
```

```yaml
      - variation: light_color
        enabled: True
        targets: [DefaultLightA, DefaultLightB, DefaultLightD]
        color_range: [[0.0, 0.0, 0.0], [0.5, 0.5, 0.5]]
        seed: ${env.seed}

      - variation: table_texture
        enabled: True
        seed: ${env.seed}

      - variation: table_color
        enabled: True
        color_range: [[0.25, 0.25, 0.25], [1.0, 1.0, 1.0]]
        seed: ${env.seed}

      - variation: background_texture
        enabled: True
        seed: ${env.seed}

      - variation: distractor_object
        enabled: True
        targets: [spawn_boundary0]
        num_objects: 2
        seed: ${env.seed}

      - variation: distractor_object
        enabled: True
        targets: [spawn_boundary1]
        num_objects: 2
        seed: 43

      - variation: camera_pose
        enabled: True
        targets: [cam_front,
                  cam_over_shoulder_left,
                  cam_over_shoulder_right]
        euler_range: [[-0.05, -0.05, -0.05], [0.05, 0.05, 0.05]]
        position_range: [[-0.1, -0.1, -0.1], [0.1, 0.1, 0.1]]
        seed: ${env.seed}
```

Fig. 2: Sample of a yaml configuration file for THE COLOSSEUM for one task. This configuration file controls the application of each perturbation_factors for this task. One or more factors can be applied at the same time in one task instance, as compatible.

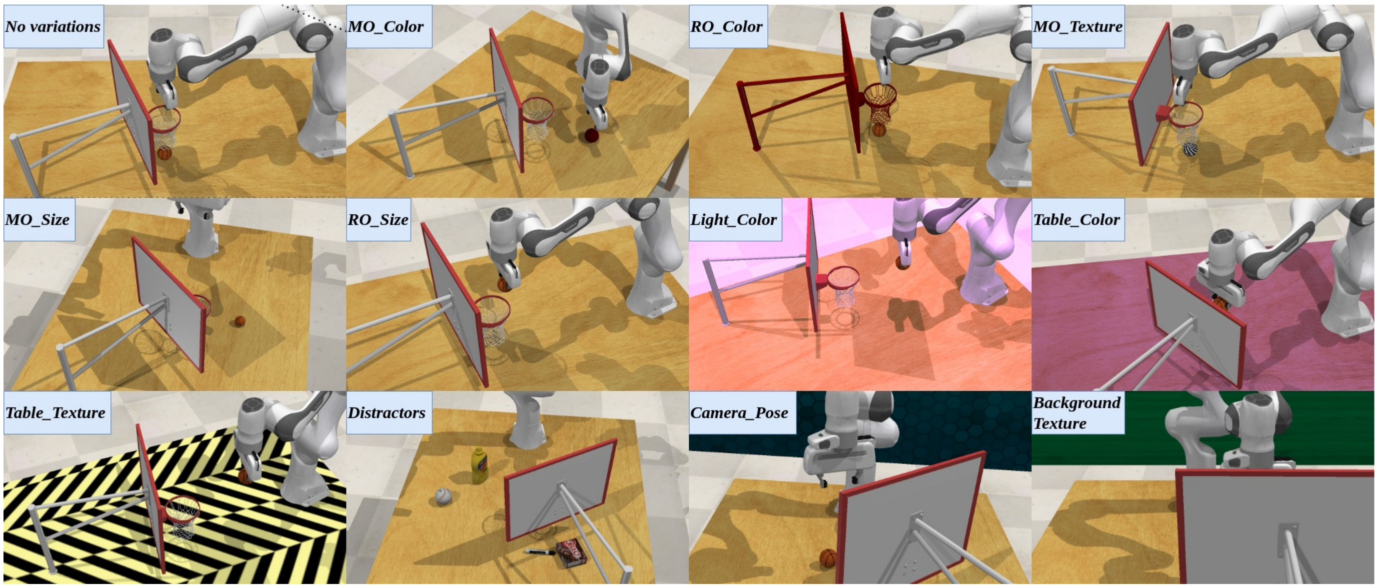


Fig. 3: Perturbations for the basketball_in_hoop task

TABLE I: Real-world ablation study

Combination of perturbations	**The Colosseumś perturbations**	**Realistic real-world scenarios**	**Correlation**
Distractors + MO_Size	[30,30,20,30,10]	[10,0,10,10,30]	0.75
Distractors+ Light_Color	[70,60,70,70,50], Cell 2	[50,40,50,40,50]	0.01
Light_Color+Table_Texture+Distractor+MO_Size	[30,40,40,40,30]	[30,10,10,20,30]	0.83

Variation / Task	MO	RO	MO_Color	MO_Size	MO_Texture	RO_Color	RO_Size	RO_Texture	Object Mass
	-	-	discrete	continuous	discrete	discrete	continuous	discrete	continuous
`basketball_in_hoop`	ball	hoop	20	$[0.75, 1.25]$	213	20	$[0.75, 1.15]$	-	-
`close_box`	box	-	20	$[0.75, 1.15]$	-	-	-	-	-
`close_laptop_lid`	laptop	-	20	$[0.75, 1.00]$	-	-	-	-	-
`empty_dishwasher`	dishwasher	plate	20	$[0.80, 1.00]$	-	20	$[0.80, 1.00]$	213	-
`get_ice_from_fridge`	cup	fridge	20	$[0.75, 1.25]$	213	20	$[0.75, 1.00]$	-	-
`hockey`	stick	ball	20	$[0.95, 1.05]$	-	20	$[0.75, 1.25]$	213	$[0.1, 0.5]$
`meat_on_grill`	meat	grill	20	$[0.65, 1.15]$	-	20	-	-	-
`move_hanger`	hanger	pole	20	-	-	20	-	-	-
`wipe_desk`	sponge	beans	20	$[0.75, 1.25]$	213	20	-	-	$[1.0, 5.0]$
`open_drawer`	drawer	-	20	$[0.75, 1.00]$	-	-	-	-	-
`slide_block_to_target`	block	-	20	-	213	-	-	-	$[1.0, 15.0]$
`reach_and_drag`	stick	block	20	$[0.80, 1.10]$	213	20	$[0.50, 1.00]$	213	$[0.5, 2.5]$
`put_money_in_safe`	money	safe	20	$[0.50, 1.00]$	213	20	-	213	-
`place_wine_at_rack_location`	bottle	shelve	20	$[0.85, 1.15]$	-	20	$[0.85, 1.15]$	213	-
`insert_onto_square_peg`	peg	spokes	20	$[1.00, 1.50]$	-	20	$[0.85, 1.15]$	213	-
`stack_cups`	cups	-	20	$[0.75, 1.25]$	213	-	-	-	-
`turn_oven_on`	knobs	-	20	$[0.50, 1.50]$	-	-	-	-	-
`straighten_rope`	rope	-	20	-	213	-	-	-	-
`setup_chess`	chess pieces	board	20	$[0.75, 1.25]$	213	20	-	-	-
`scoop_with_spatula`	spatula	block	20	$[0.75, 1.25]$	213	20	$[0.75, 1.50]$	213	$[1.0, 5.0]$

TABLE II: Summary of tasks and their `perturbation_factors`. The table specifies when a certain factor is applied to a certain task and its corresponding parameters.

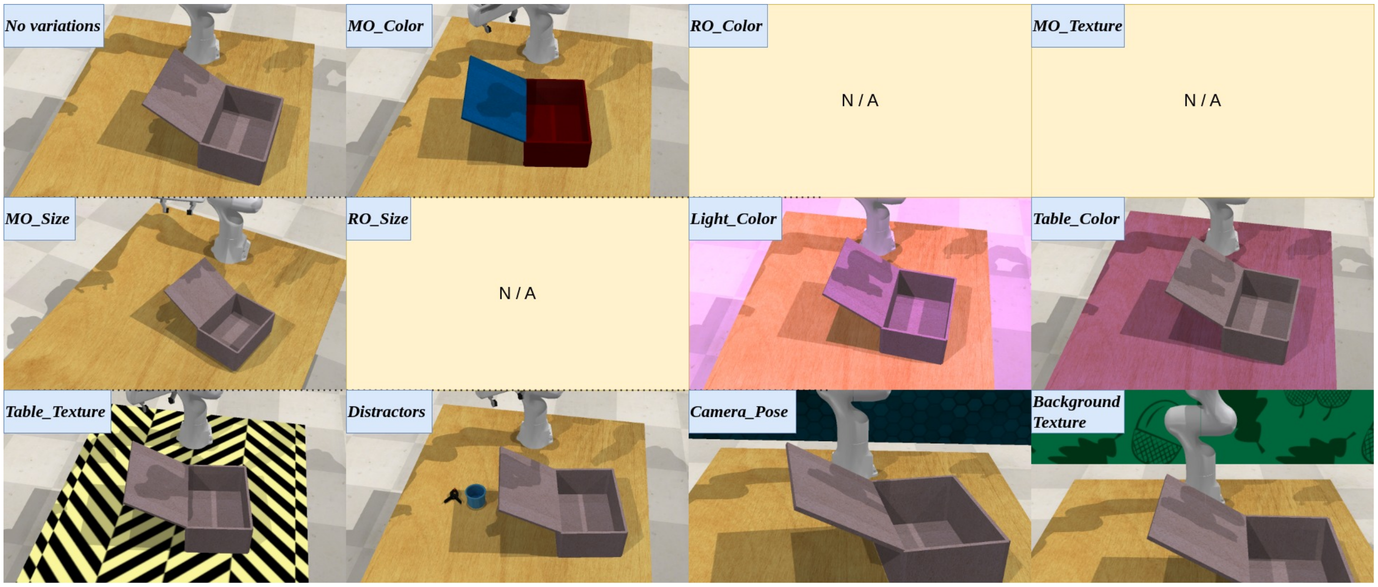


Fig. 4: Perturbations for the `close_box` task

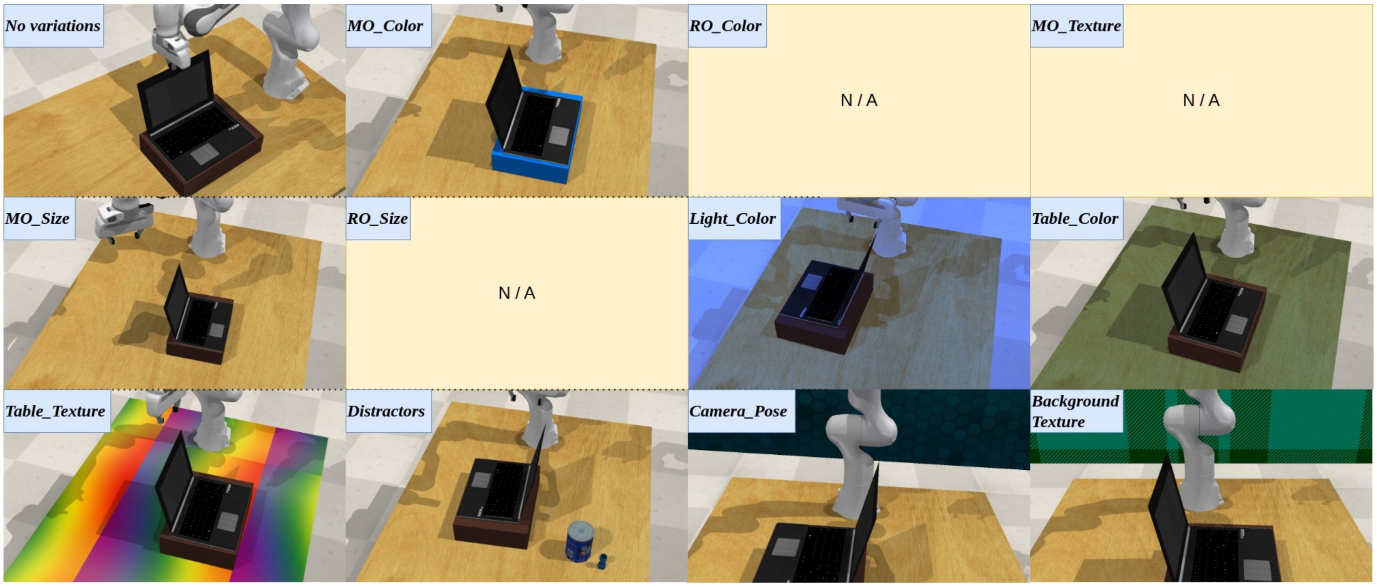


Fig. 5: Perturbations for the `close_laptop_lid` task

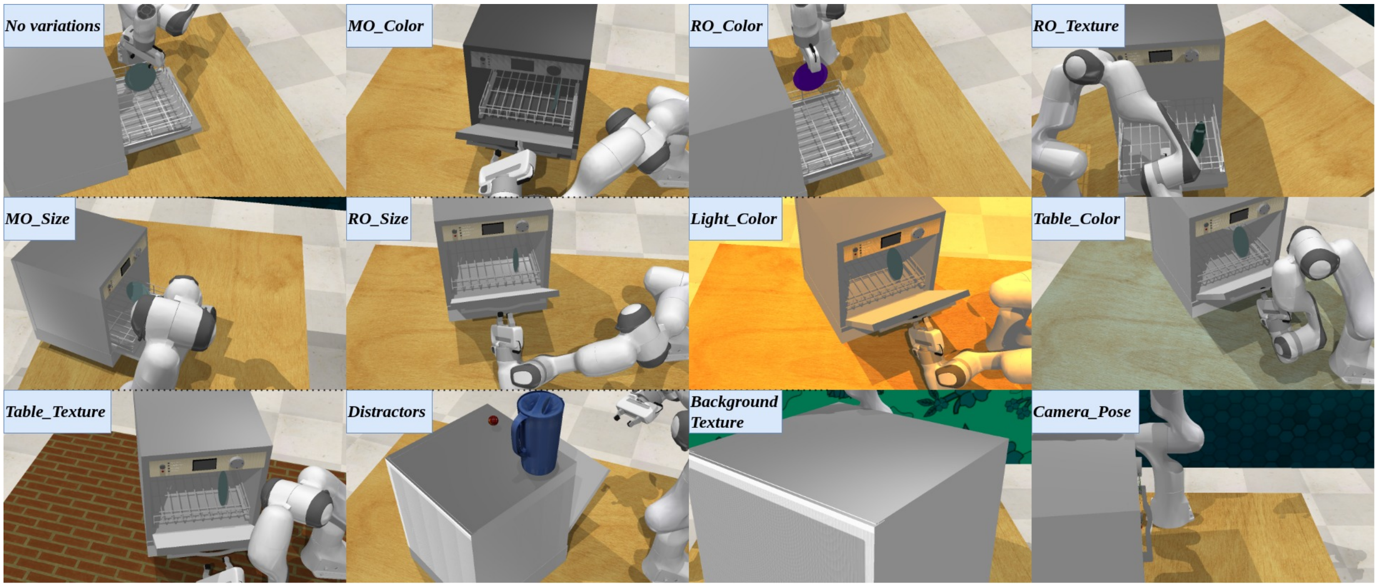


Fig. 6: Perturbations for the `empty_dishwasher` task

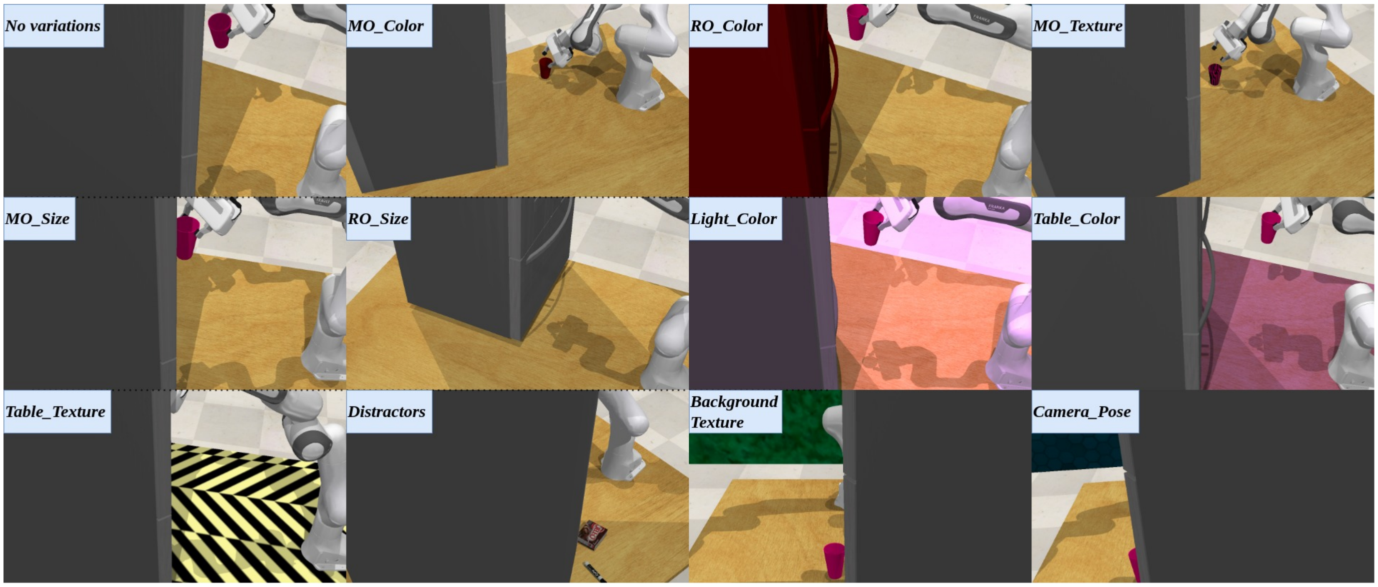


Fig. 7: Perturbations for the `get_ice_from_fridge` task

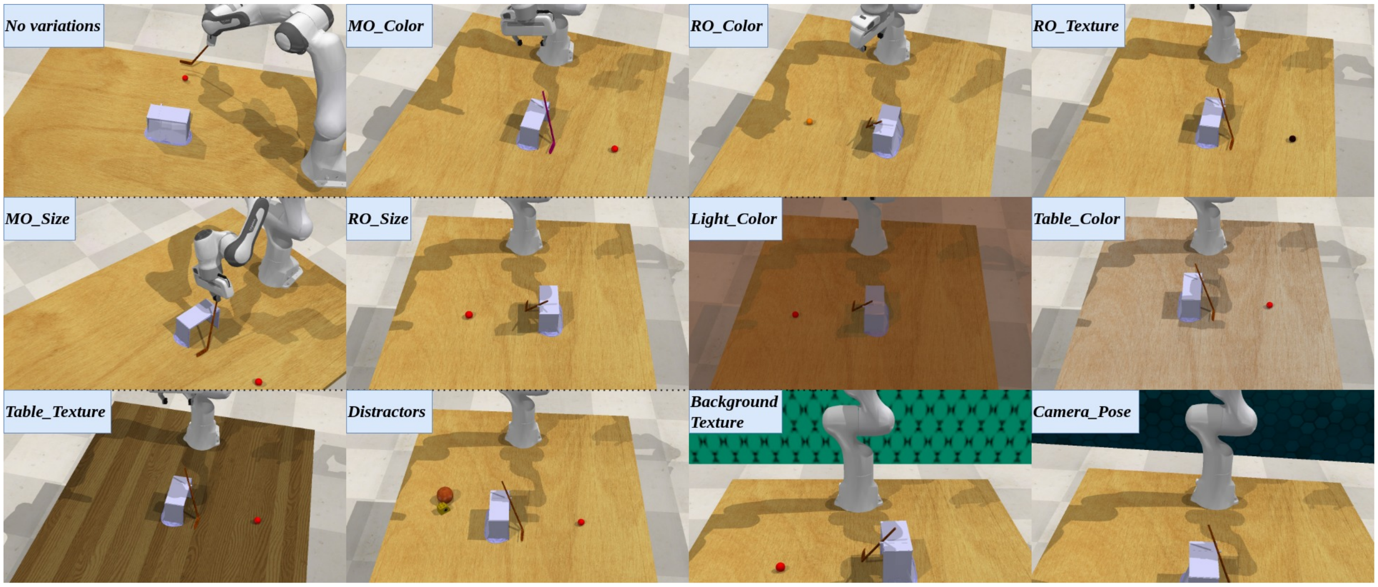


Fig. 8: Perturbations for the `hockey` task

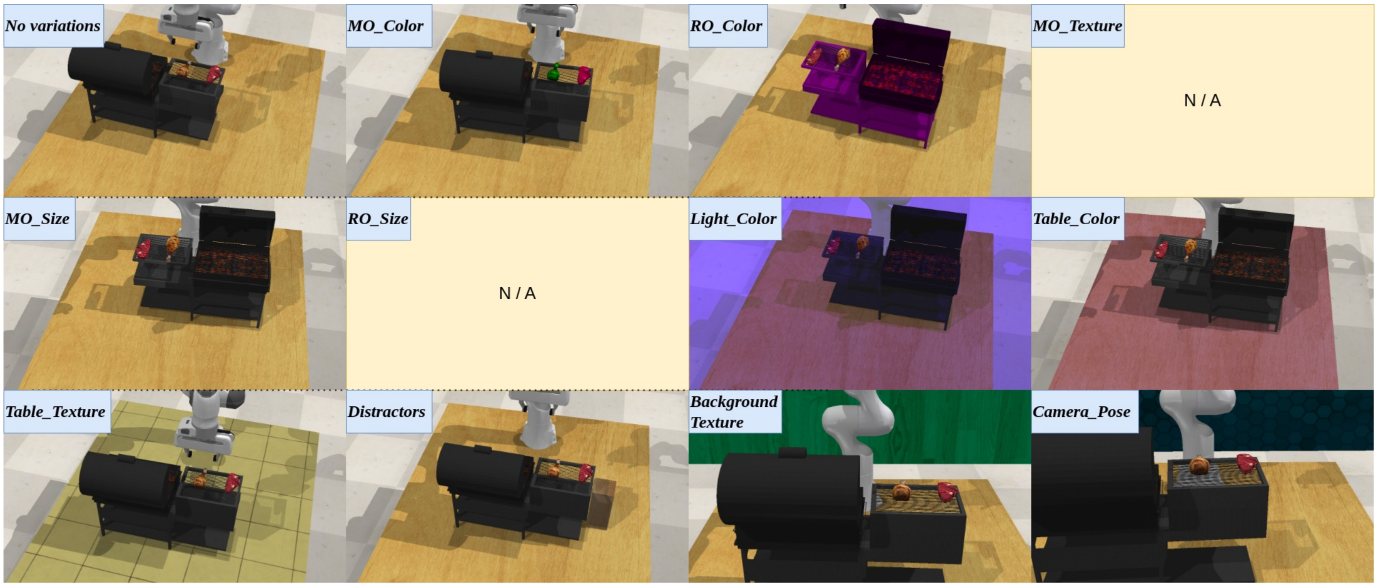


Fig. 9: Perturbations for the `meat_on_grill` task

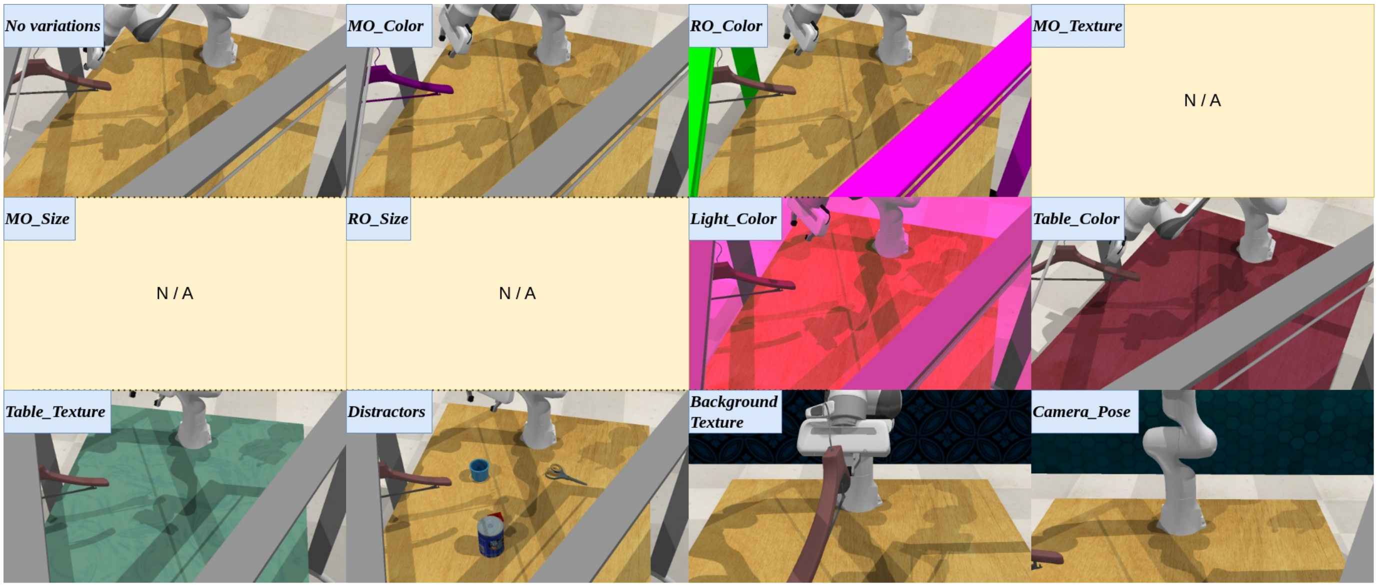


Fig. 10: Perturbations for the `move_hanger` task

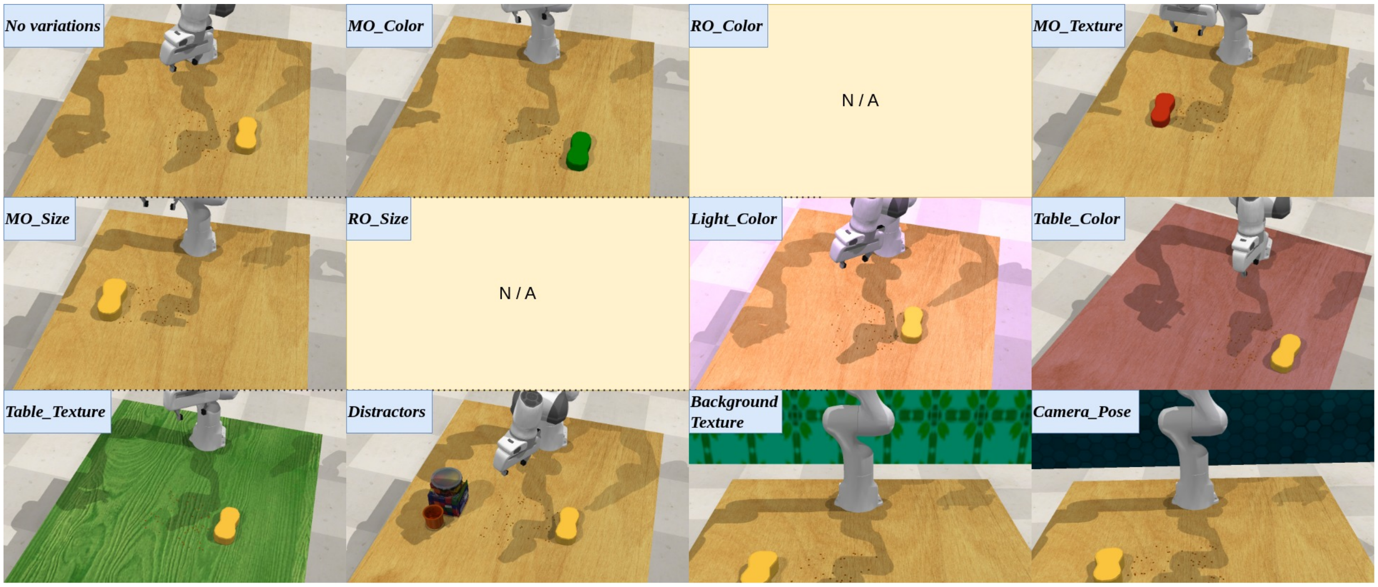


Fig. 11: Perturbations for the `wipe_desk` task

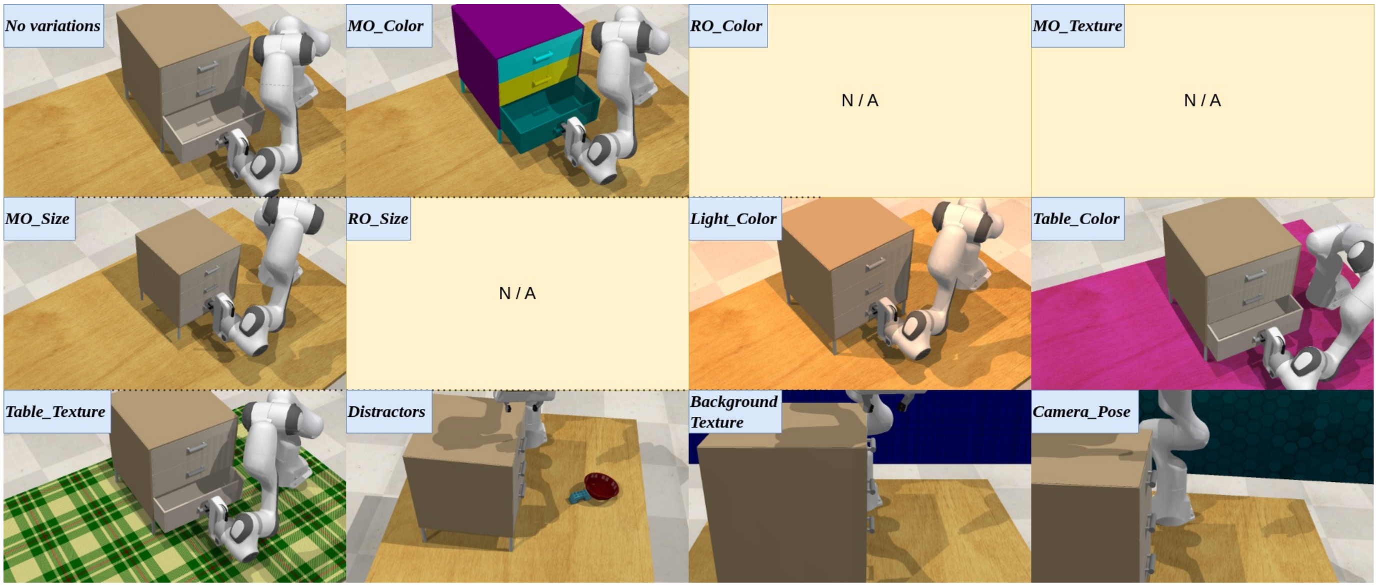


Fig. 12: Perturbations for the `open_drawer` task

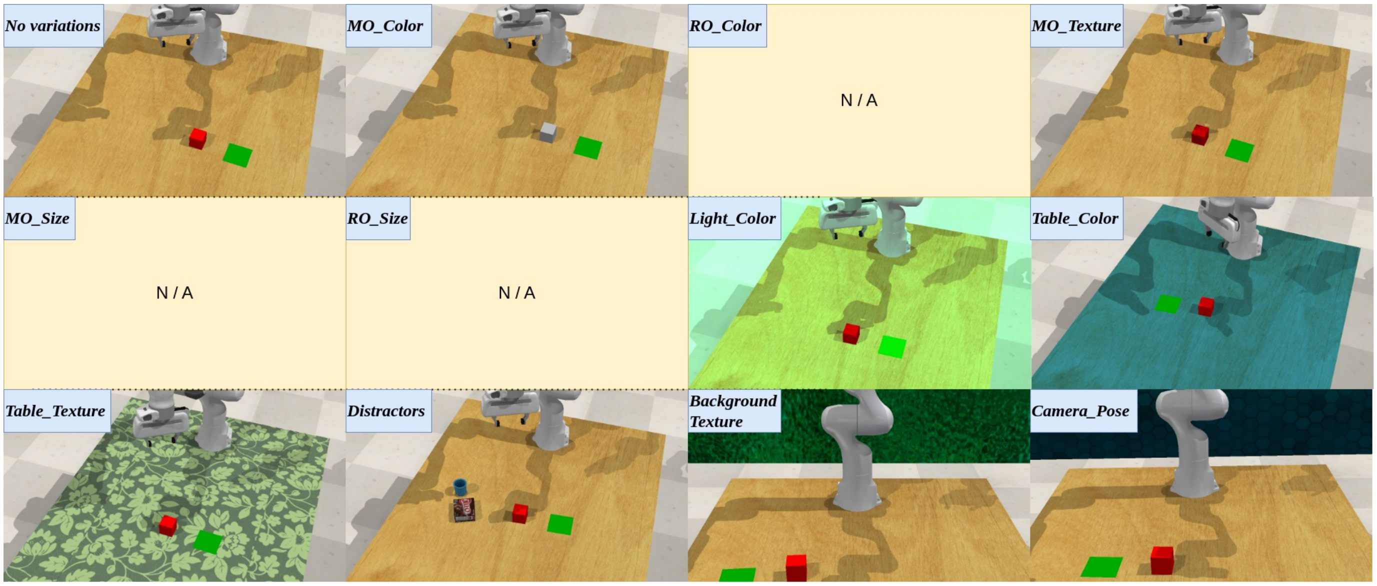


Fig. 13: Perturbations for the `slide_block_to_target` task

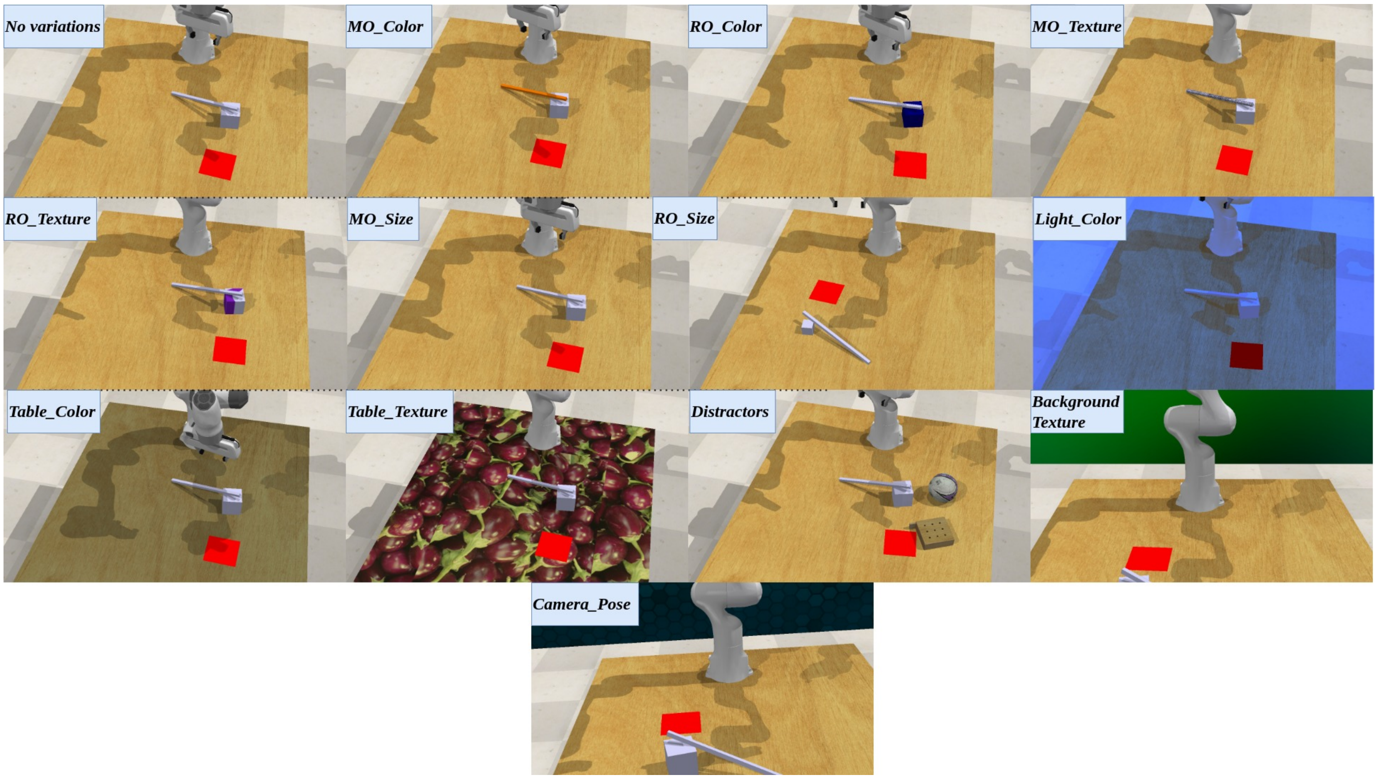


Fig. 14: Perturbations for the `reach_and_drag` task

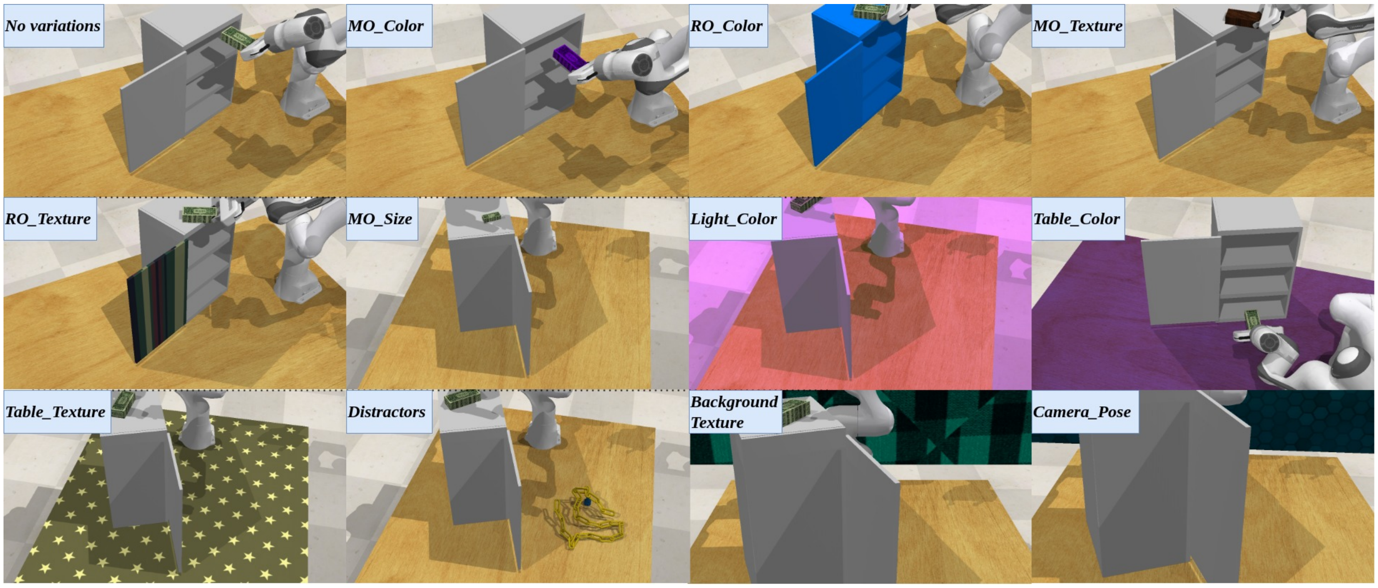


Fig. 15: Perturbations for the `put_money_in_safe` task

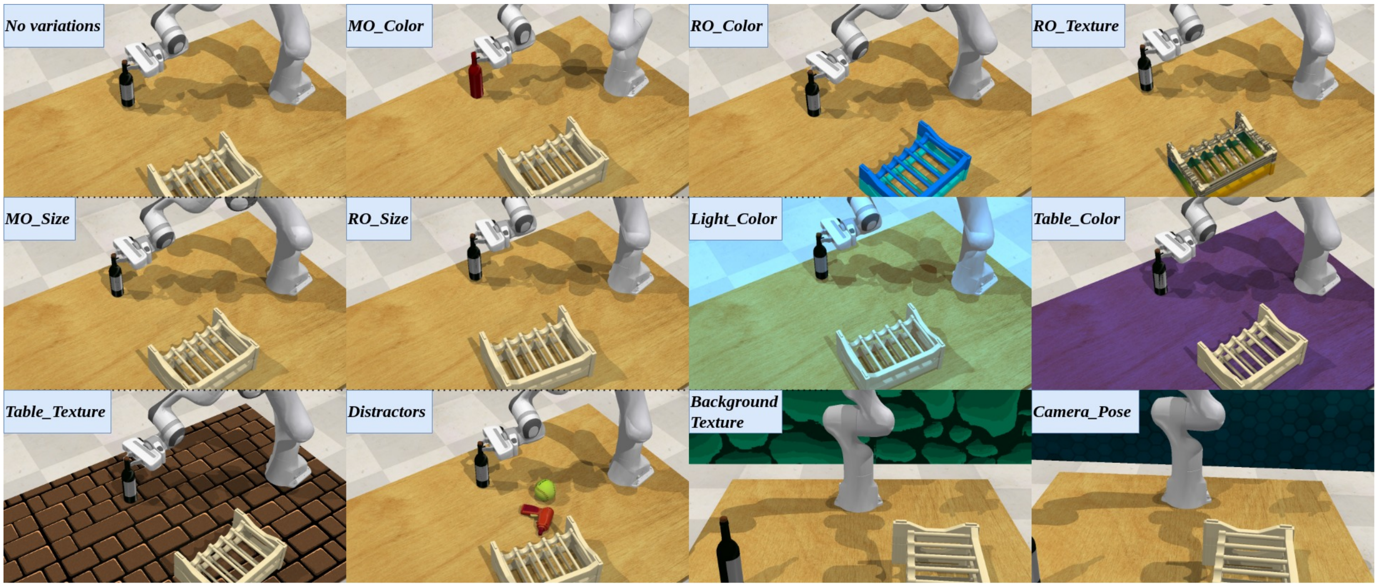


Fig. 16: Perturbations for the `place_wine_at_rack_location` task

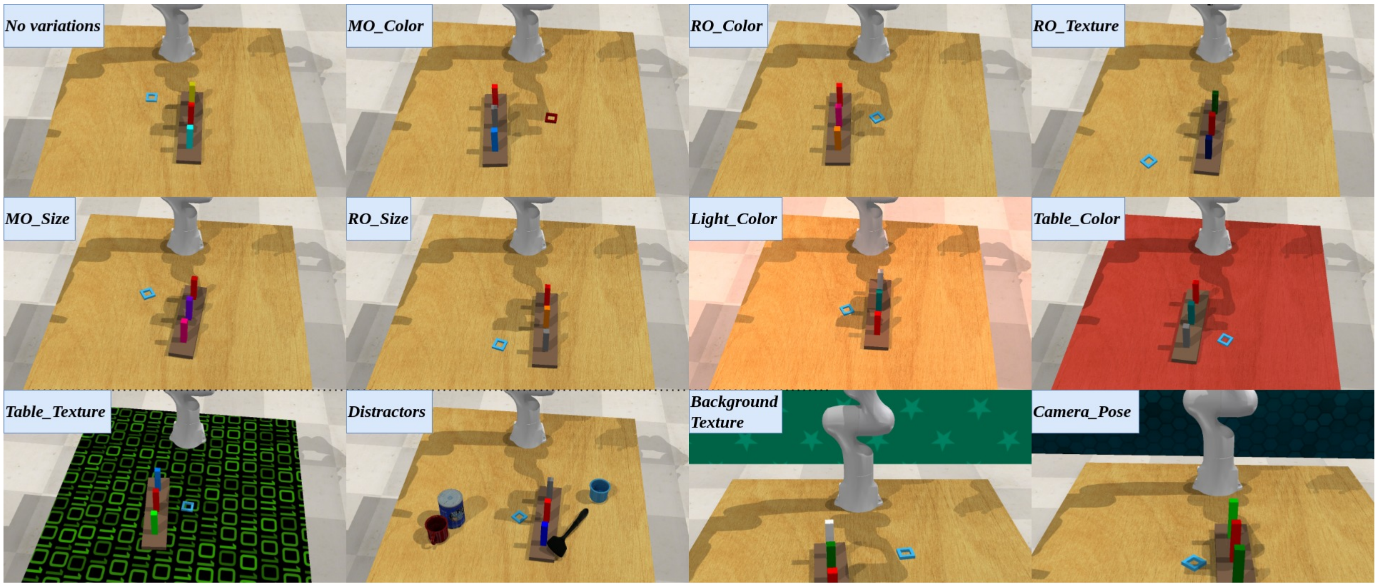


Fig. 17: Perturbations for the `insert_onto_square_peg_location` task

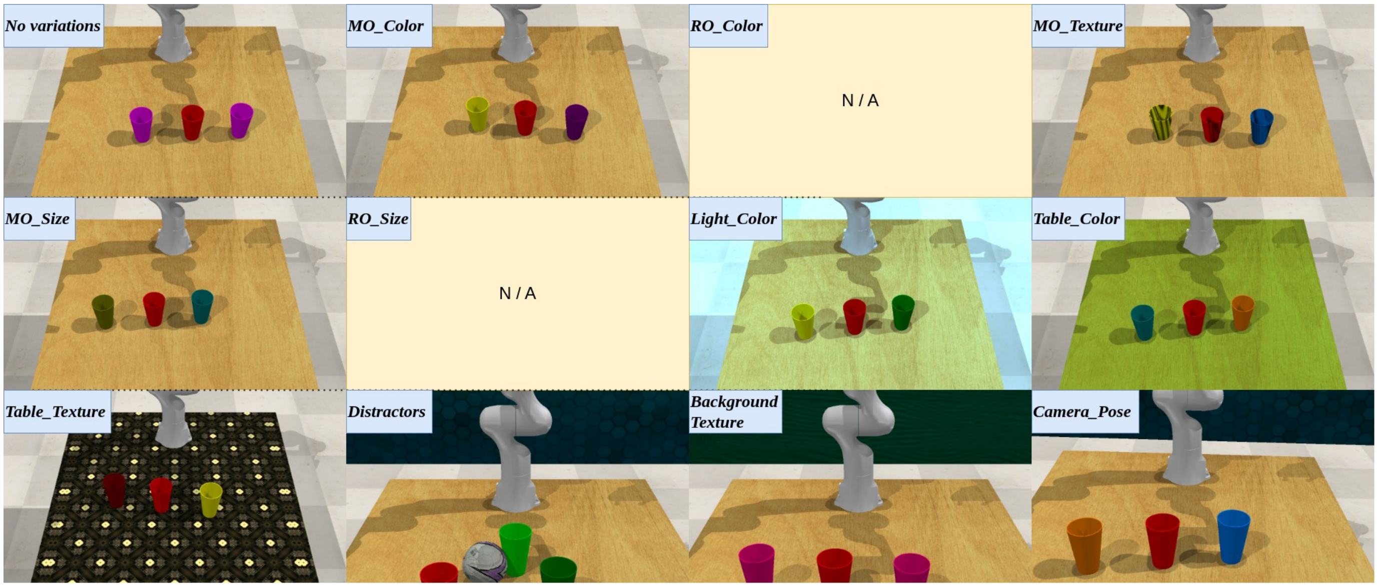


Fig. 18: Perturbations for the `stack_cups` task

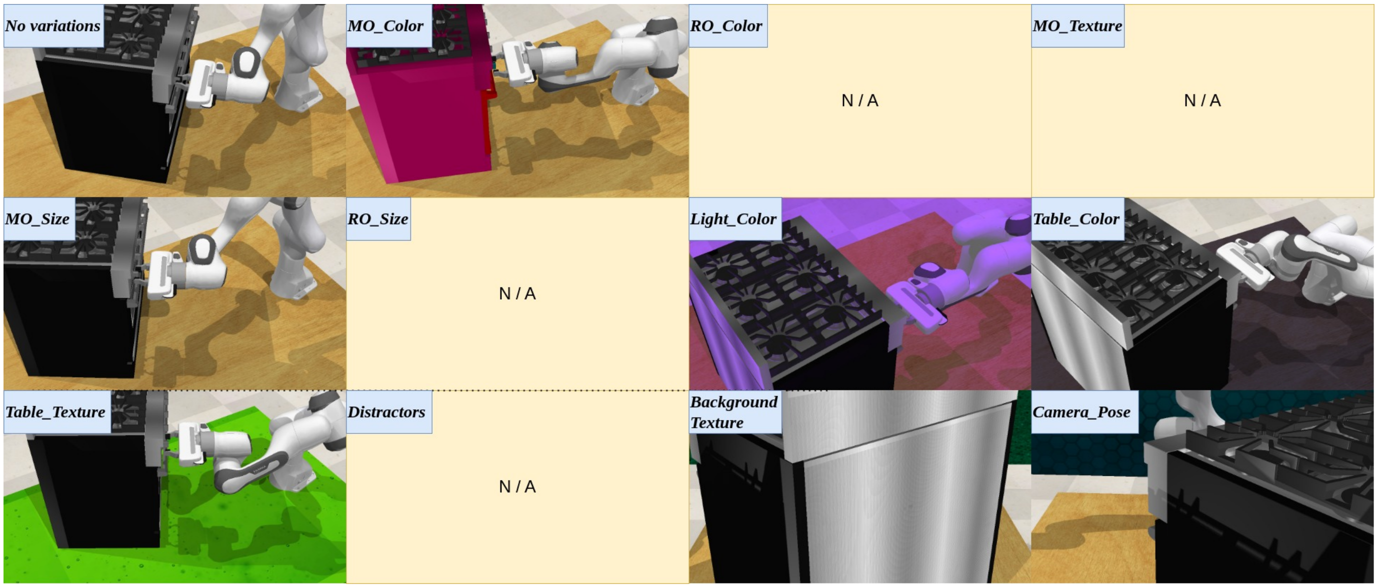


Fig. 19: Perturbations for the `turn_oven_on` task

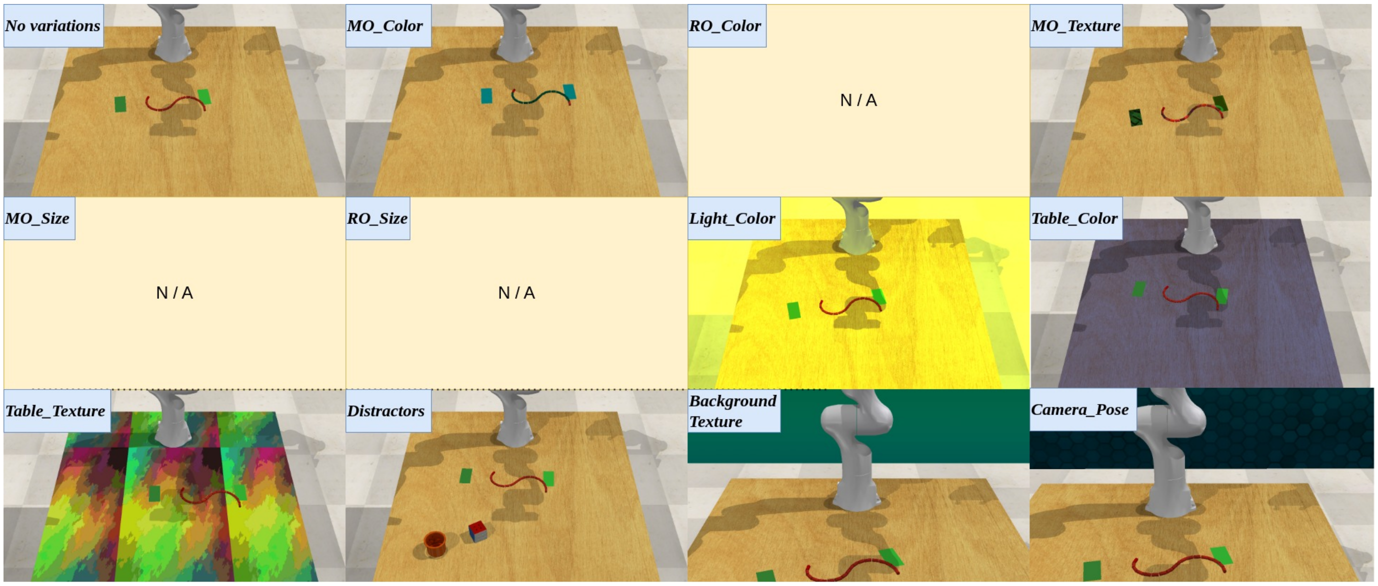


Fig. 20: Perturbations for the `straighten_rope` task

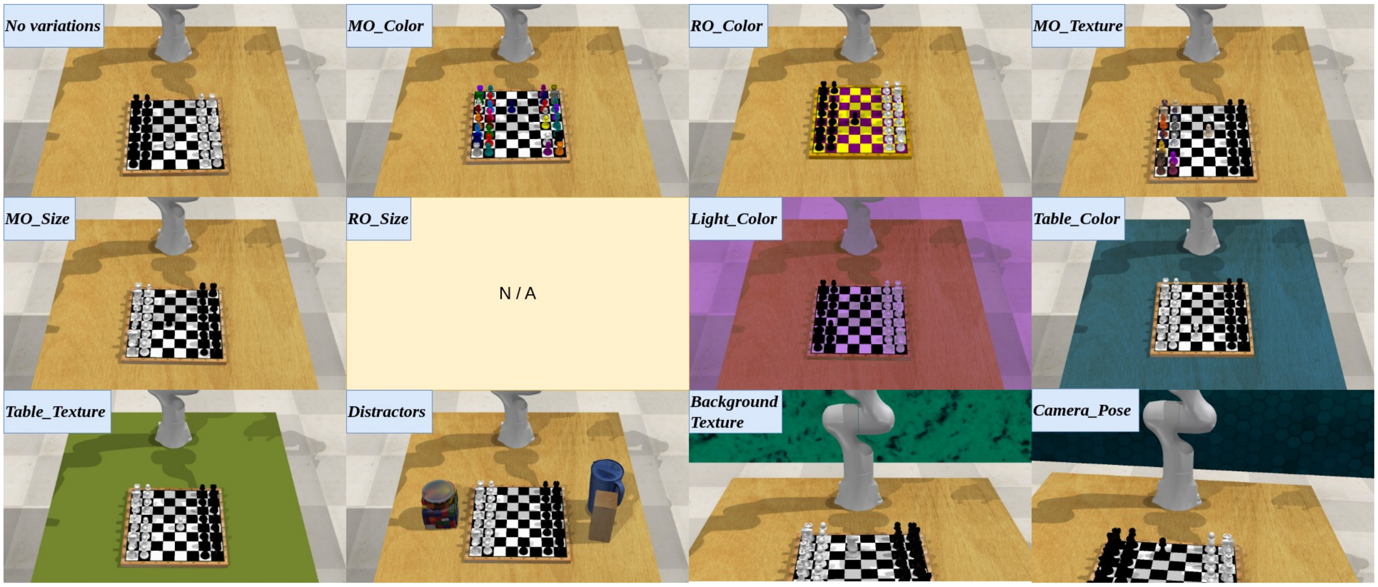


Fig. 21: Perturbations for the `setup_chess` task

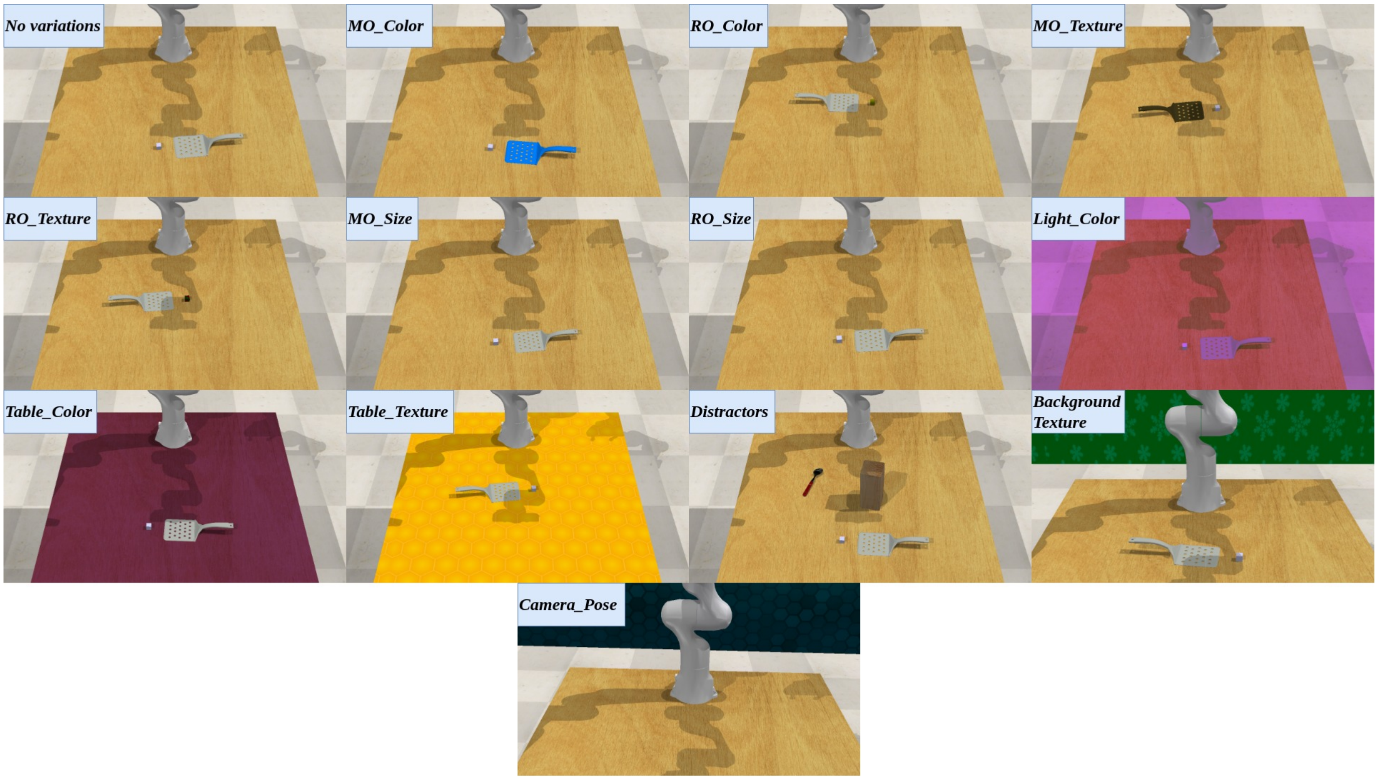


Fig. 22: Perturbations for the `scoop_with_spatula` task

Task Name	No variations	All variations	MO_Color	RO_Color	MO_Texture	RO_Texture	MO_Size	RO_Size	Light-color	Table-Color	Table-texture	Distractor	Backgrond-texture	RLBench variations	Camera pose	Object Friction	Object Mass
basketball_in_hoop	100	0	100	72	100	-	100	74	92	84	76	48	96	100	96	-	-
close_box	65	0	28	-	-	-	40	-	50	15	28	30	50	52	64	-	-
close_laptop_lid	96	80	80	-	-	-	100	-	92	88	80	88	100	100	96	-	-
empty_dishwasher	0	0	0	0	-	0	0	0	0	0	0	0	0	4	0	-	-
get_ice_from_fridge	60	4	60	60	56	-	60	68	60	76	40	72	76	76	84	-	-
hockey	0	0	0	0	-‘	0	0	0	0	0	0	0	0	0	0	0	0
meat_on_grill	92	44	64	72	-	-	92	-	64	92	60	88	80	92	84	-	-
move_hanger	0	0	0	0	-	-	-	-	0	0	0	0	0	0	0	-	-
wipe_desk	0	0	0	0	0	-	0	-	0	0	0	0	0	0	0	0	0
open_drawer	28	0	0	-	-	-	0	-	16	80	32	28	8	68	76	-	-
slide_block_to_target	24	0	4	-	16	-	68	-	20	8	4	12	32	32	0	-	-
reach_and_drag	36	0	20	12	4	8	40	8	12	12	8	0	20	64	20	12	24
put_money_in_safe	32	0	32	16	44	28	20	-	28	12	12	20	20	44	20	-	-
place_wine_at_rack_location	0	0	0	0	-	0	8	12	8	0	4	0	4	8	8	-	-
insert_onto_square_peg	4	0	0	4	-	4	0	8	8	4	0	8	4	28	0	-	-
stack_cups	8	0	12	-	0	-	0	-	0	16	0	0	4	0	8	-	-
turn_oven_on	24	8	20	-	-	-	40	-	40	40	48	40	36	32	40	-	-
straighten_rope	0	0	0	-	0	-	-	-	0	0	0	0	0	4	14	-	-
setup_chess	44	8	28	76	44	-	0	-	56	64	64	48	68	16	60	-	-
scoop_with_spatula	76	0	32	68	24	84	72	64	36	16	8	60	72	68	56	64	64

TABLE III: Results for `PerAct` for various perturbations

Task Name	No variations	All variations	MO_Color	RO_Color	MO_Texture	RO_Texture	MO_Size	RO_Size	Light-color	Table-Color	Table-texture	Distractor	Background-texture	RLBench variations	Camera pose	Object Friction	Object Mass
basketball_in_hoop	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	-	-
close_box	32	0	0	-	-	-	24	-	-	12	0	0	12	24	8	-	-
close_laptop_lid	6	4	4	-	-	-	4	-	8	12	0	20	4	4	4	-	-
empty_dishwasher	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
get_ice_from_fridge	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	-	-
hockey	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0
meat_on_grill	0	0	0	0	-	-	0	-	0	0	0	0	0	0	0	-	-
move_hanger	4	0	0	0	-	-	-	-	0	0	0	0	0	0	0	-	-
wipe_desk	0	0	0	0	0	-	0	-	0	0	0	0	0	0	0	0	0
open_drawer	0	0	0	-	-	-	0	-	0	0	0	0	0	0	0	-	-
slide_block_to_target	0	0	0	-	0	-	0	-	0	0	0	0	0	0	0	-	-
reach_and_drag	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
put_money_in_safe	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	-	-
place_wine_at_rack_location	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
insert_onto_square_peg	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
stack_cups	0	0	0	-	0	-	0	-	0	0	0	0	0	0	0	-	-
turn_oven_on	12	8	4	-	-	-	4	-	12	4	4	12	8	12	4	-	-
straighten_rope	0	0	0	-	0	-	-	-	0	0	0	0	0	0	0	-	-
setup_chess	4	0	0	0	0	-	0	-	0	0	0	0	0	0	0	-	-
scoop_with_spatula	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE IV: Results for `R3M` for various perturbations

Task Name	No variations	All variations	MO_Color	RO_Color	MO_Texture	RO_Texture	MO_Size	RO_Size	Light-color	Table-Color	Table-texture	Distractor	Background-texture	RLBench variations	Camera pose	Object Friction	Object Mass
basketball_in_hoop	4	0	4	0	4	-	0	4	4	4	0	4	0	0	0	-	-
close_box	40	8	12	-	-	-	60	-	8	8	12	8	36	24	12	-	-
close_laptop_lid	8	0	8	-	-	-	0	-	16	4	4	40	0	4	20	-	-
empty_dishwasher	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
get_ice_from_fridge	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	-	-
hockey	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0
meat_on_grill	10	4	0	0	-	-	0	-	0	0	0	12	0	0	4	-	-
move_hanger	0	0	0	0	-	-	-	-	0	0	0	0	0	0	0	-	-
wipe_desk	0	0	0	0	0	-	0	-	0	0	0	0	0	0	0	0	0
open_drawer	0	0	0	-	-	-	0	-	0	0	0	0	0	0	0	-	-
slide_block_to_target	0	0	0	-	0	-	0	-	0	0	0	0	0	0	0	-	-
reach_and_drag	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
put_money_in_safe	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	-	-
place_wine_at_rack_location	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
insert_onto_square_peg	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
stack_cups	0	0	0	-	0	-	0	-	0	0	0	0	0	0	0	-	-
turn_oven_on	6	4	0	-	-	-	20	-	4	16	4	12	8	12	16	-	-
straighten_rope	0	0	0	-	0	-	-	-	0	0	0	0	0	0	0	-	-
setup_chess	0	0	0	0	0	-	0	-	0	0	0	0	0	0	0	-	-
scoop_with_spatula	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE V: Results for MVP for various perturbations

Task Name	No variations	All variations	MO_Color	RO_Color	MO_Texture	RO_Texture	MO_Size	RO_Size	Light-color	Table-Color	Table-texture	Distractor	Background-texture	RLBench variations	Camera pose	Object Friction	Object Mass
basketball_in_hoop	84	4	92	4	68	-	80	84	32	28	88	16	88	100	68	-	-
close_box	80	36	8	-	-	-	84	-	96	56	80	80	84	92	68	-	-
close_laptop_lid	52	24	80	-	-	-	24	-	36	48	64	20	68	68	56	-	-
empty_dishwasher	0	4	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
get_ice_from_fridge	80	0	68	44	84	-	56	88	72	60	68	40	84	68	80	-	-
hockey	4	0	0	0	-	0	0	4	28	36	0	0	0	0	0	0	0
meat_on_grill	12	40	16	56	-	-	8	-	28	12	4	12	8	76	4	-	-
move_hanger	80	0	0	96	-	-	-	-	0	0	100	8	84	84	0	-	-
wipe_desk	0	0	0	0	0	-	0	-	0	0	0	0	0	0	0	0	0
open_drawer	64	0	0	-	-	-	72	-	68	64	68	52	52	72	72	-	-
slide_block_to_target	0	0	0	-	0	-	0	-	0	0	0	0	0	72	0	-	-
reach_and_drag	84	0	24	52	88	88	92	0	72	52	88	4	88	76	80	88	88
put_money_in_safe	44	0	68	0	44	28	52	-	16	16	16	32	36	60	64	-	-
place_wine_at_rack_location	60	12	72	40	-	72	36	64	88	88	60	32	52	56	72	-	-
insert_onto_square_peg	4	0	0	16	-	12	24	4	8	16	20	4	4	8	8	-	-
stack_cups	0	0	12	-	12	-	0	-	40	12	24	0	16	24	20	-	-
turn_oven_on	88	8	40	-	-	-	28	-	52	36	80	72	92	80	80	-	-
straighten_rope	32	0	20	-	48	-	-	-	0	28	52	4	92	40	68	-	-
setup_chess	24	0	4	-	24	-	16	-	0	4	8	0	4	4	20	-	-
scoop_with_spatula	80	0	16	68	80	88	64	80	44	44	84	0	76	88	84	72	68

TABLE VI: Results for RVT for various perturbations

Task Name	No variations	All variations	MO_Color	RO_Color	MO_Texture	RO_Texture	MO_Size	RO_Size	Light-color	Table-Color	Table-texture	Distractor	Background-texture	RLBench variations	Camera pose	Object Friction	Object Mass
basketball_in_hoop	32	40	56	48	32	-	60	52	40	40	60	61	44	44	56	-	-
close_box	0	0	0	-	-	-	0	-	0	0	0	0	0	0	0	-	-
close_laptop_lid	0	0	0	-	-	-	0	-	0	0	0	0	0	0	0	-	-
empty_dishwasher	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
get_ice_from_fridge	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	-	-
hockey	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0
meat_on_grill	0	0	0	0	-	-	0	-	0	0	0	0	0	0	0	-	-
move_hanger	0	0	0	0	-	-	-	-	0	0	0	0	0	0	0	-	-
wipe_desk	0	0	0	0	0	-	0	-	0	0	0	0	0	0	0	0	0
open_drawer	0	0	0	-	-	-	0	-	0	0	0	0	0	0	0	-	-
slide_block_to_target	76	80	72	-	70	-	-	-	60	64	84	76	88	80	68	-	-
reach_and_drag	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
put_money_in_safe	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	-	-
place_wine_at_rack_location	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
insert_onto_square_peg	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	-	-
stack_cups	0	0	0	-	0	-	0	-	0	0	0	0	0	0	0	-	-
turn_oven_on	0	0	0	-	-	-	0	-	0	0	0	0	0	0	0	-	-
straighten_rope	0	0	0	-	0	-	-	-	0	0	0	0	0	0	0	-	-
setup_chess	0	0	0	-	0	-	0	-	0	0	0	0	0	0	0	-	-
scoop_with_spatula	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE VII: Results for `Voxposer` for various perturbations

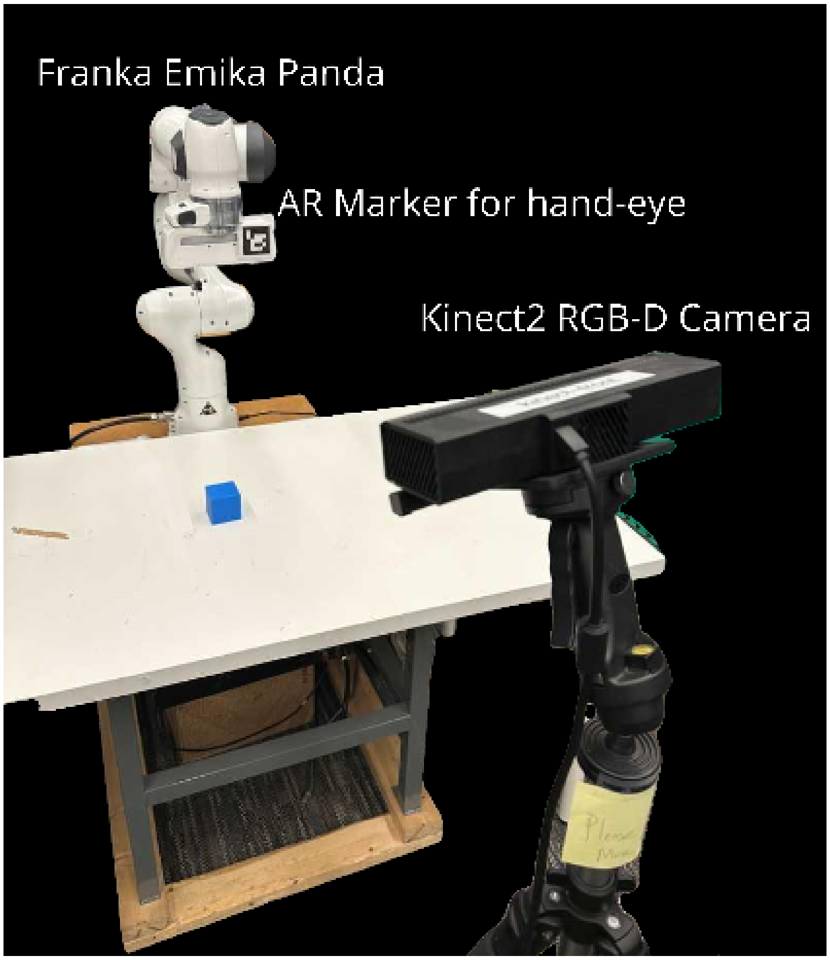


Fig. 23: **Real-Robot Setup with Kinect-2 and Franka Panda.**

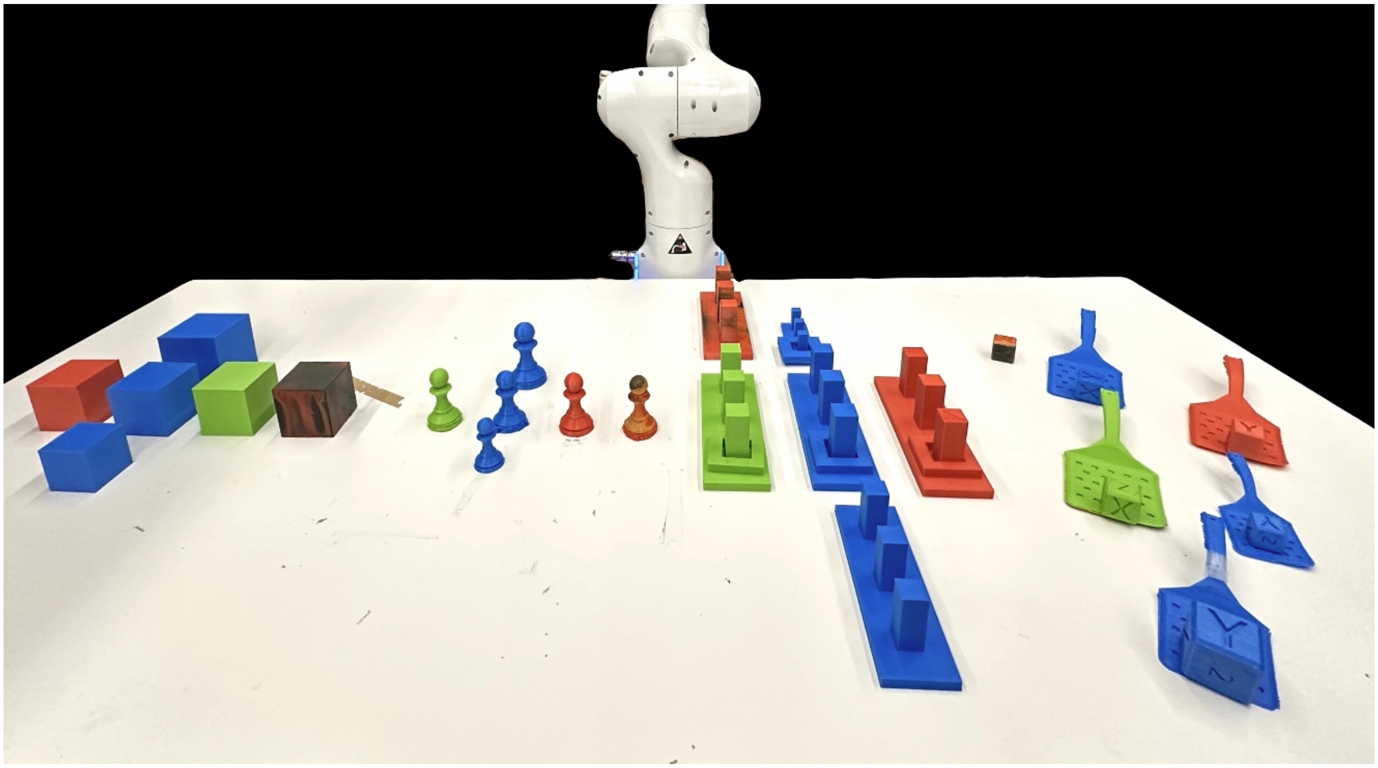


Fig. 24: **3D print-outs of all the assets for the real-world tasks.**