Closed Loop Interactive Embodied Reasoning for Robot Manipulation
Supplementary material

Michal Nazarczuk, Jan Kristof Behrens, Karla Stepanova, Matej Hoffmann and Krystian Mikolajczyk

A. *Weight measurements details*

To perform weight measurements in our experiment, we rely on the external force reported by libfranka. We assume that the only external force is incurred by the objects held in the hand, i.e., that the robot is not in collision with the environment. To avoid errors due to model inaccuracies, we measure the force in the z-direction twice: once with the empty gripper and once with the object lifted to the same robot configuration. The difference divided by the constant g proved to be a reliable indicator of the weight sufficient for our applications. If more accurate numbers are needed, we could use a digital scale, but this would have no effect on the reasoning side.

For simplicity, we assume that for each considered property, we have a suited (primitive) measurement procedure that interacts with the objects and produces an estimate of the property. It is true that we could, for example, from the same interaction estimate the object's inertia or the center of gravity. Reusing the data would possibly lead to shorter action sequences for complex queries, but this is not in the scope of this paper.

Attributing a measurement to several objects is currently not implemented. It is theoretically possible to formulate constraints on sets of properties to encode and reason about combined measurements. We will consider that in our future work.

In Figure 1 we show graphs of the Force in the z direction over time for measuring the weight of a mug and a spam can. Figure 2 shows the robot in the moments of force assessment (also marked in the graphs by the vertical lines). Our method results in an estimate of the mug weight of 95.5g (ground truth 96g) and 22.2g for the spam can (ground truth 24g), ground truth shown in Figure 3. For a measurement with the empty gripper, we observed an estimated weight of -2 g. This shows that we can distinguish the weight of all objects in the presented tasks and also reach a surprisingly good absolute precision.

B. *An example of the scene graph*

In order to better explain the details of our approach, in Figure 4 we show an example of the execution of the query that includes notation for program execution, action planning, and updates of the scene graph.

C. *Stiffness measurements details*

In addition to our benchmark, we propose an extension with another invisible physical property - stiffness. We provide an example of the task execution in the accompanying

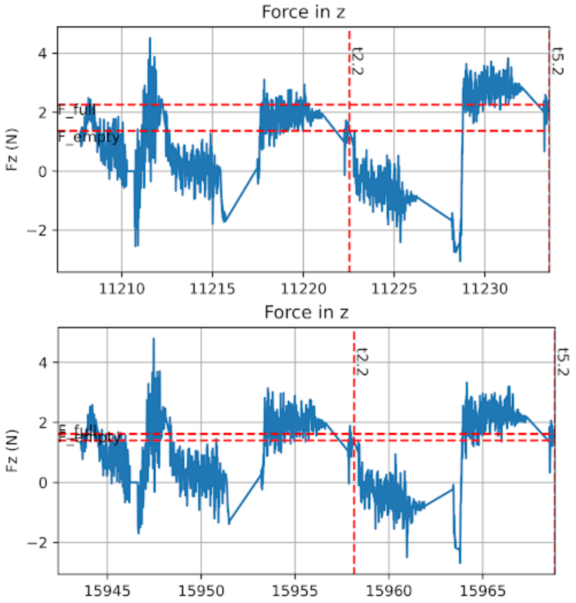


Fig. 1. Graphs of the force in the z direction for weighing the mug (top) and the spam can (bottom)

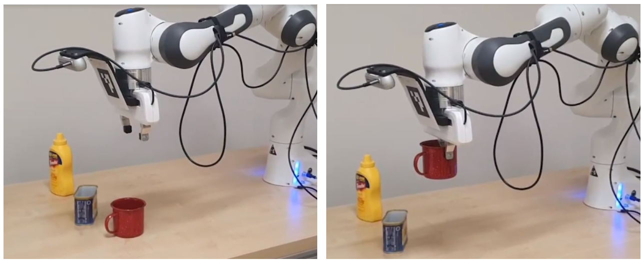


Fig. 2. Robot configurations corresponding to weight estimation.

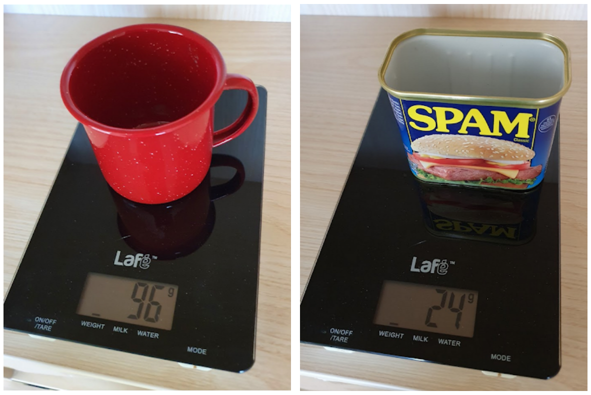
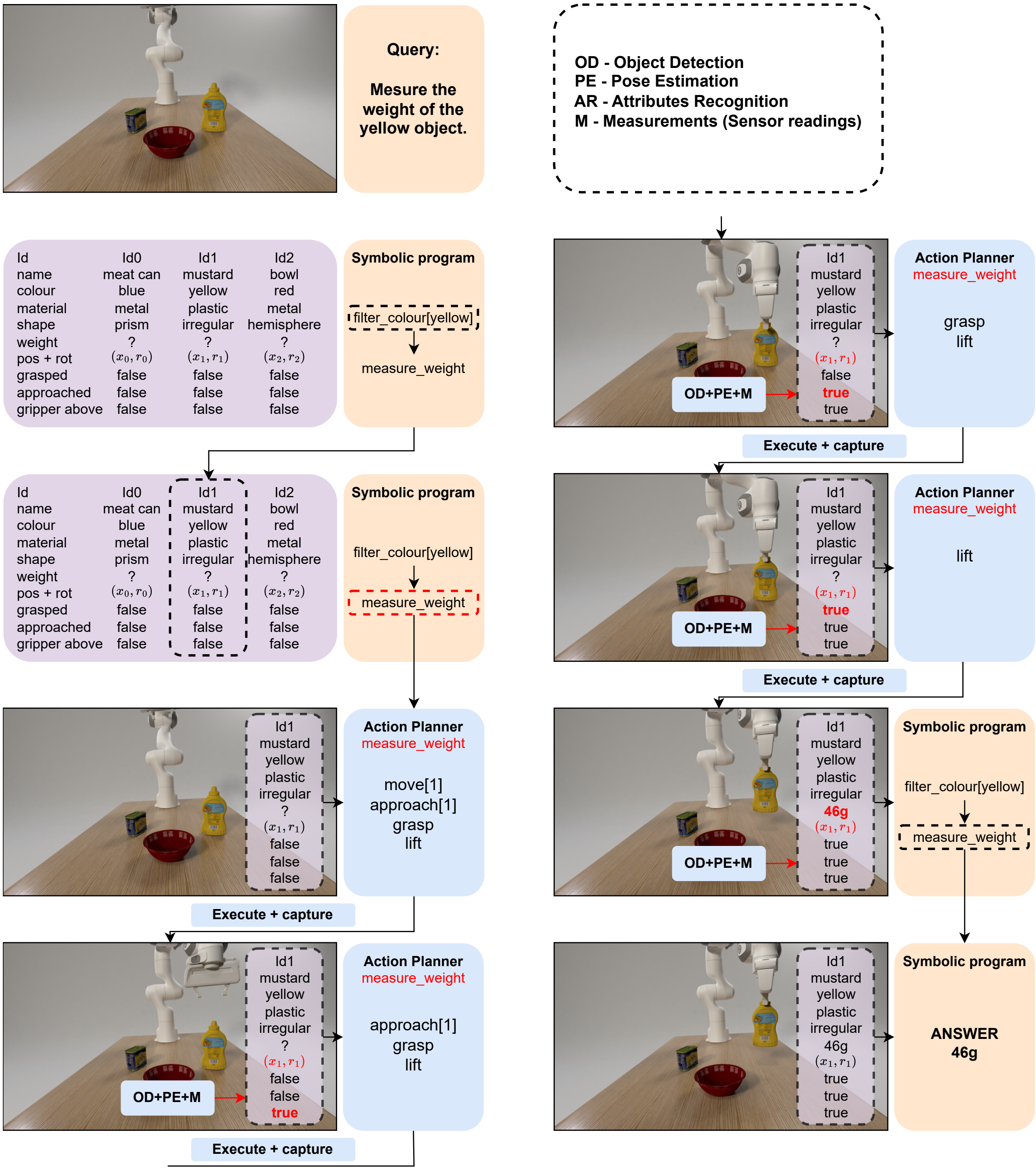


Fig. 3. Ground truth weight measurements for mug (left) and spam can (right).



Id	Id0	Id1	Id2
name	meat can	mustard	bowl
colour	blue	yellow	red
material	metal	plastic	metal
shape	prism	irregular	hemisphere
weight	?	?	?
pos + rot	(x_0, r_0)	(x_1, r_1)	(x_2, r_2)
grasped	false	false	false
approached	false	false	false
gripper above	false	false	false

Id	Id0	Id1	Id2
name	meat can	mustard	bowl
colour	blue	yellow	red
material	metal	plastic	metal
shape	prism	irregular	hemisphere
weight	?	?	?
pos + rot	(x_0, r_0)	(x_1, r_1)	(x_2, r_2)
grasped	false	false	false
approached	false	false	false
gripper above	false	false	false

Fig. 4. An example of task execution by CLIER approach.

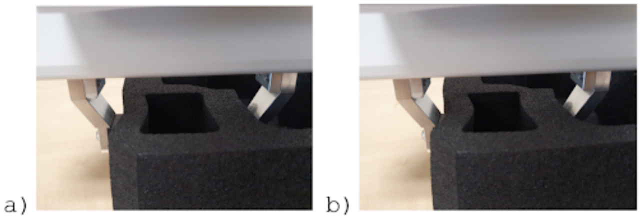


Fig. 5. Stiffness measurement procedure.

video (see the last part of the supplementary video for the execution of task “Pick up the stiffest metal object”). The stiffness measurement procedure that we implemented in our physical setup utilises the ability of the robot to close the gripper while not exceeding a given force limit. First, we close the gripper with a minimal force $F_1 = 0.1N$ and read the aperture d_1 of the gripper (see Fig. 5a). Then we open the gripper slightly and close it again with a higher force $F_2 = 50N$ and read the aperture d_2 (see Fig. 5b). The stiffness estimate is then given as $k = \frac{F_2 - F_1}{|d_2 - d_1|}$.

D. Horizon length comparison

We believe that our action planning is robust to the horizon due to the design of our system. Specifically, the transformer is fed with a scene graph and a subgoal target (obtained using the seq2seq symbolic program generator) as the input to predict the next primitive action. Then, after every action, a new action plan is generated. Therefore, the history of the interactions (memory) is stored within the scene graph, and reasoning is always performed from the current state. In Figure6, we show the graph of the success rate of the experiment with respect to the ground truth number of steps for the given instruction. We see a slight decrease in accuracy with the increasing length of the action sequence. However, we do not attribute that to the reasoning module itself (which on its own performs at 86.7%, and we did not observe a correlation between step index and occurrence of errors). Qualitative error analysis suggests that the decrease in accuracy can be attributed to errors in scene graph updates, specifically, pose regression errors that lead to missed grasping.

E. Error codes details

This section, covers the list of possible outcome codes for the CLIER framework, along with an explanation of possible shortcomings of the tested model. The codes include:

- Correct answer - the answer to the query is correct (last program item is non-actionable);
- Task success - the task was executed correctly (last program item is actionable);
- Task failure - last action was executed but the task was not resolved correctly;
- Execution error - action executor did not exit with a success status within the timeout limit (usually caused by infeasible configuration or physics conflict, *e.g.* errors in position estimation leading to excessive forces acting on objects);
- Loop detected - a sequence of repeating actions was detected (usually caused by incorrectly estimating the pose and repeating the sequence of approaching and grasping the object);
- Physics error - caused when an object is thrown out of the workspace due to excessive forces during manipulation;
- Program error - error in the inference of the instruction;
- Recognition error - errors in the attributes recognition leading to incorrect scene graph creation;
- Output error - incorrect output of the program (possibly caused by incorrect scene update);
- Scene inconsistency - high discrepancy in the recognition of attributes and poses in the consecutive frames.

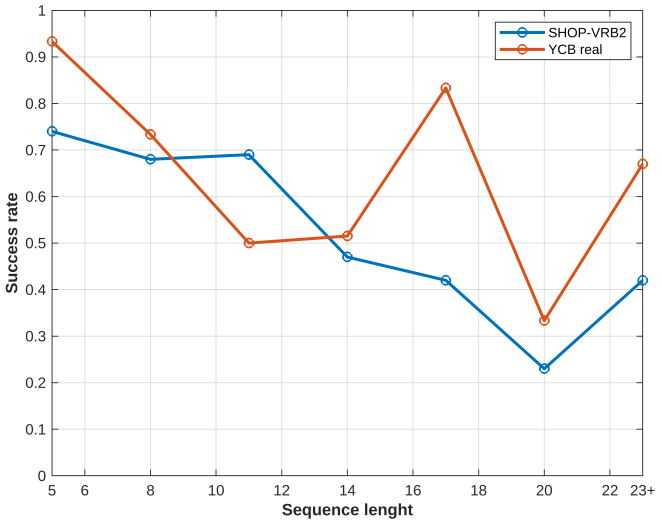


Fig. 6. Success rate of CLIER prediction with respect to ground truth task length.

F. Details on primitive actions

In this section, we present some more details on how primitive actions are translated into motion paths. Note that the actions take the estimated scene graph as the argument (that includes estimated positions, orientations, and heuristic assessing whether the object is in the hand of the gripper).

1) **`move`***:* The action `move` can take either part of the table, or the target object as the argument. Given the object as the target of the action, the final position of the end effector is calculated such that either the centre of the currently held object or the middle of the end effector are directly above the centre of the target object. The height of the target position is estimated based on the object bounding boxes to keep a given clearance value. The trajectory of the movement is currently based on the bounding boxes of all objects in the scene. Any motion planning algorithm can be implemented in that place according to the needs.

When a part of the table is passed as an argument, movement target is found at a convenient free space when a gripper is currently holding an object. A set of coordinates are tested, spiralling from the middle of the target zone (*e.g.* left part of the table), and the position is considered suitable if no bounding boxes are intersecting.

2) ***lift**, **lower**:* Actions of lifting and lowering the manipulator are simply upward and downward movements that consider given clearances of bounding boxes with respect to the tabletop.

3) ***open_gripper**, **close_gripper**:* Actions controlling the gripper consider a current gripper state, and control the gripper to keep closing until further closure is not viable, or open till fully opened position.

4) ***approach**:* We consider a few main cases of grasping objects: grasping of cylinders, grasping by the edge, grasping by the handle, grasping of boxes.

For grasping of the cylinder, an object's orientation is taken into account. If the object is withing the tolerance of standing in the upright position, a top grasp is considered. If the bounding box of the object signifies bigger dimensions than grippers capability, or one of the dimensions in xy-plane is significantly different to other, the grasp is reduced to upright edge grasp. Otherwise, an object is grasped from top, at its axis of symmetry, avoiding gripper rotation. Similarly, if the object is considered to be laying on the tabletop, a top grasp over the object is performed if viable.

For the edge grasping, a similar check for the object position (standing/laying) is performed. In both cases the object is grasped by the edge. Gripper is positioned on top of the edge for laying object, and for standing, perpendicular to the bounding box longer dimension (in xy-plane) in order to avoid possibility of handles interfering.

For the grasping by the handle, it is assumed that handle appears at the longer side of the objects (according to bounding box). A grasp is performed with a certain offset from the boundary of object (scaling with the size of the object).

For the grasping of boxes, the smaller dimension of the bounding box of the object in xy-plane is used to align the gripper along shorter axis. Thereafter, the grasp is performed in the middle of the bounding box (in xy-plane) at a given depth.

G. Scene graph update procedure

The scene graph contains the categorical attributes for every object, along with their positions and orientations. At each update, we align a newly recognised set of objects to the current scene based on matching their attributes (followed by similarity of poses for mismatch cases). Additionally, we use simple geometrical heuristics on the positions of the objects and the position of the end effector of the manipulator in order to define the following attributes: whether the object is in the gripper hand, whether the object is within the feasible grasp, whether the object is raised above table, whether the gripper is placed directly above the object.