

rc_kitchen_table_case10

RoboCerebra | goal: Prepare a simple dish by combining tomato sauce and macaroni and cheese

9 annotated steps + annotation inventory + methodology



Subgoal 1 [manipulate]: Open the white cabinet on top side
 | target='white cabinet' (green GT mask,
 focus_kind=container, 1837px) | related_objects: white
 cabinet | frames {'start': 0, 'end': 346}



Subgoal 2 [grasp]: Pick up the tomato sauce from the
 kitchen table | target='tomato sauce' (green GT mask,
 focus_kind=object, 135px) | related_objects: tomato
 sauce, kitchen table | frames {'start': 346, 'end': 673}



Subgoal 3 [release]: Place down the tomato sauce into the
 frypan | target='tomato sauce' (green GT mask,
 focus_kind=object, 295px) | related_objects: tomato
 sauce, frypan | frames {'start': 673, 'end': 816}



Subgoal 4 [release]: Place the tomato sauce on the kitchen table. | target='tomato sauce' (green GT mask, focus_kind=object, 386px) | related_objects: tomato sauce, kitchen table | frames {'start': 816, 'end': 1012}



Subgoal 5 [grasp]: Pick up the macaroni and cheese from the kitchen table | target='macaroni and cheese' (green GT mask, focus_kind=object, 591px) | related_objects: macaroni and cheese, kitchen table | frames {'start': 1012, 'end': 1300}



Subgoal 6 [release]: Place down the macaroni and cheese into the frypan | target='macaroni and cheese' (green GT mask, focus_kind=object, 413px) | related_objects: macaroni and cheese, frypan | frames {'start': 1300, 'end': 1450}



Subgoal 7 [release]: Place the macaroni and cheese on the kitchen table. | target='macaroni and cheese' (green GT mask, focus_kind=object, 839px) | related_objects: macaroni and cheese, kitchen table | frames {'start': 1450, 'end': 1656}



Subgoal 8 [grasp]: Pick up the wine bottle from the kitchen table | target='wine bottle' (green GT mask, focus_kind=object, 146px) | related_objects: wine bottle, kitchen table | frames {'start': 1656, 'end': 2051}



Subgoal 9 [manipulate]: Pour out the contents of the wine bottle into the red coffee mug | target='red coffee mug' (green GT mask, focus_kind=object, 912px) | related_objects: wine bottle, red coffee mug | frames {'start': 2051, 'end': 2447}

ANNOTATION INVENTORY -- rc_kitchen_table_case10

long-horizon goal: Prepare a simple dish by combining tomato sauce and macaroni and cheese in a chefmate 8" frypan and pour wine from a wine bottle into a red coffee mug as a beverage.

1. Subgoal decomposition . 9 subtasks (LLM-decomposed, human-verified) -> subtask + phase
[TEMPORAL]

2. related_objects per-subtask target object list (from task_description) [SEMANTIC]

3. timestep per-subtask frame range (start,end) [TEMPORAL]

4. Focus GT mask sim instance-seg of the target object, per subgoal -> green overlay
[SPATIAL, engine-exact]

focus_kind / mask_px .. how the target was matched to a sim instance + mask area.

5. Forced-memory markers . keyframes=9 ; onset_occluded_steps=1 -> steps whose target is
already occluded at onset (answer needs MEMORY, not current frame) -- the held-out test
condition.

6. states / actions full recorded MuJoCo trajectory (enables the re-sim that yields the
masks).

How these annotations were produced (RoboCerebra + our pipeline):

- RoboCerebra ships RGB video + (states, actions) + a task_description (high-level instruction LLM-decomposed

into a subtask sequence, each with a frame range 'timestep' and its 'related_objects'). Human-verified.

- Forced-memory structure: meta.json marks 'keyframes' and 'onset_occluded_steps' -- subgoals whose target

is already occluded at the step onset, so the answer must come from MEMORY, not the current frame.

- Focus MASK (green): we RE-SIMULATE each trajectory in the LIBERO/MuJoCo engine (set every recorded mujoco

state) and read the simulator's GROUND-TRUTH instance segmentation for the subgoal's target object

(matched by name). Engine-exact, obtained by replay -- not hand-drawn. focus_kind = how it was matched;

mask_px = mask area. RoboCerebra does NOT ship masks.

- Sim render -> exact masks but synthetic appearance (visual sim2real gap).