APPENDIX

IX. RoCoBench

A. Overview

RoCoBench is built with MuJoCo [38] physics engine. The authors would like to thank the various related open-source efforts that greatly assisted the development of RoCoBench tasks: DMControl [39], Menagerie [5], and MuJoCo object assets from [7]. The sections below provide a detailed documentation for each of the 6 simulated collaboration tasks.

B. Task: Sweep Floor

Task Description. 2 Robots bring a dustpan and a broom to opposite sides of each cube to sweep it up, then the robot holding dustpan dumps cubes into a trash bin.

Agent Capability. Two robots stand on opposite sides of the table:

1) UR5E with robotiq gripper ('Alice'): holds a dustpan
2) Franka Panda ('Bob'): holds a broom

Observation Space. 1) cube locations: a. on table; b. inside dustpan; c. inside trash bin; 2) robot status: 3D gripper locations

Available Robot Skills. 1) MOVE [target]: target can only be a cube; 2) SWEEP [target]: moves the groom so it pushes the target into dustpan; 3) WAIT; 4) DUMP: dump dustpan over the top of trash bin.

C. Task: Make Sandwich

Task Description. 2 Robots make a sandwich together, each having access to a different set of ingredients. They must select the required items and take turns to stack them in the correct order.

Agent Capability. Two robots stand on opposite sides of the table:

Observation Space 1) the robot's own gripper state (either empty or holding an object); 2) food items on the robot's own side of the table and on the cutting board.

Available Robot Skills. 1) PICK [object]; 2) PUT [object] on [target]; WAIT

D. Task: Sort Cubes

Task Description. 3 Robots sort 3 cubes onto their corresponding panels. The robots must stay within their respective reach range, and help each other to move a cube closer.

Agent Capability. Three robots each responsible for one area on the table

1) UR5E with robotiq gripper ('Alice'): must put blue square on panel2, can only reach: panel1, panel2, panel3.
2) Franka Panda ('Bob'): must put pink polygon on panel4, can only reach: panel3, panel4, panel5.
3) UR5E with suction gripper ('Chad'): must put yellow trapezoid on panel6, can only reach: panel5, panel6, panel7.

Observation Space 1) the robot's own goal, 2) locations of each cube.

Available Robot Skills. 1) PICK [object] PLACE [panelX]; 2) WAIT

E. Task: Pack Grocery

Task Description. 2 Robots pack a set of grocery items from the table into a bin. The objects are in close proximity and robots must coordinate their paths to avoid collision.

Agent Capability. Two robots on opposite sides of table

1) UR5E with robotiq gripper ('Alice'): can pick and place any object on the table
2) Franka Panda ('Bob'): can pick and place any object on the table

Observation Space 1) robots' gripper locations, 2) locations of each object, 3) locations of all slots in the bin.

Available Robot Skills. (must include task-space waypoints) 1) PICK [object] PATH [path]; 2) PLACE [object] [target] PATH [path]

F. Task: Move Rope

Task Description. 2 robots lift a rope together over a wall and place it into a groove. They must coordinate their grippers to avoid collision.

Agent Capability. Two robots on opposite sides of table

1) UR5E with robotiq gripper ('Alice'): can pick and place any end of the rope within its reach
2) Franka Panda ('Bob'): can pick and place any end of the rope within its reach

Observation Space 1) robots' gripper locations, 2) locations of rope's front and end back ends; 3) locations of corners of the obstacle wall; 4) locations of left and right ends of the groove slot.

Available Robot Skills. (must include task-space waypoints) 1) PICK [object] PATH [path]; 2) PLACE [object] [target] PATH [path]

G. Task: Arrange Cabinet

Task Description. 3 robots, two of them each hold one side of the cabinet door open, while the third robot takes the cups out and place them onto the correct coasters.

Agent Capability. Three robots, one on left side of the table, two on right side of table

1) UR5E with robotiq gripper ('Alice'): stands on left side, can only reach left cabinet door
2) Franka Panda ('Bob'): : stands on right side, can only reach right cabinet door
3) UR5E with suction gripper ('Chad'):: stands on right side, can reach right cabinet door and cups and mugs inside the cabinet.

Observation Space 1) locations cabinet door handles; 2) each robot's reachable objects, unaware of other robot's reach range.

Available Robot Skills. 1) PICK [object]; 2) OPEN [one side of door handle]; 3) WAIT; 3) PLACE [object] [target]

X. Details on LLM Prompting

We describe our proposed method of multi-agent dialog in Algorithm 1: during each call to **PromptDialogs**, each agent speaks at least once before reaching an acton plan; and after each call to **GiveFeedback**, the proposed plan is passed through a set of validation check and optionally results in a text feedback that's used in the next round of dialog; the finalized plan is used by **MotionPlanner** to produce robot motion trajectories for execution in the environment.

To produce each agent response in a dialog, we use a separate query call to the LLM with an agent-specific prompt. The prompt provides information regarding the agent's capability, the overall task objective, past history, plan feedback, and environment observation. As a concrete example, we provide in the text box below the LLM prompt for one agent at the second time-step during one evaluation run of the Sort Cube task (some texts are omitted for readability)

Algorithm 1 Multi-agent dialog for collaboration

Require: agent $u^1, ..., u^N$, task horizon T;
Require: max number of re-plans K, max number of dialog per round M,
Require: history buffer H; feedback buffer F
  $t \leftarrow 0$
  $o\_t \leftarrow$ env.reset()
  H.empty()
  while $t < T$ **do**
    F.empty()
    while len(F)$< K$ **do**
      dialog, plan $\leftarrow$ PromptDialogs(H, F, o_t, u^n)
      plan-valid, feedback $\leftarrow$ GiveFeedback(plan)
      if plan-valid **then**
        final-plan $\leftarrow$ parsed-plan
        break
      end if
      F.append(feedback)
    end while
    if plan-valid **then**
      $\sigma\_t \leftarrow$ MotionPlanner(o_t, final-plan)
      $o\_t + 1, r\_t + 1 \leftarrow$ env.step(σ_t)
      if $r\_t + 1 > 0$ **then**
        break
      end if
    end if
    H.append(dialog)
    $t \leftarrow t + 1$
  end while

1. Agent Capability
[Action Options]
1) PICK [object name] PLACE [location] 2) WAIT
Only PICK an object if your gripper is empty. Target [location] for PLACE should be panel or a bin.
[Action Output Instruction]
You must first output 'EXECUTE ', then give **exactly** one action per robot [omitted: rest of format instruction]
2. Round History
[History]
== Round#0 ==
[Chat History]
[Alice]: [...]
[Bob]: Hello Alice and Chad, I am Bob. [omitted: rest of dialog history]
[Executed Action]
Alice: PICK pink_polygon PLACE panel3
Bob: PICK yellow_trapezoid PLACE panel5
Chad: PICK blue_square PLACE panel7
== Current Round ==
3. Task Context
7 panels on the table, ordered left to right: panel1,...,panel7. They form a straight assembly line, panel1 is closed to panel2 and farthest from panel7. You are robot Alice in front of panel2. You are collaborating with Bob, Chad to sort cubes into their target panels. The task is NOT done until all three cubes are sorted.
4. Current Observation
At current round:
blue_square is on panel7
pink_polygon is on panel3
yellow_trapezoid is on panel5
Your goal is to place blue_square on panel2, but you can only reach panel1, panel2, panel3 [omitted: rest of observation description]
5. Communication Instruction
When you respond, tell others about your goal and all constraints.[omitted: rest of communication instruction]
[Previous Chat]
[Alice]: [omitted: rest of previous dialog in current round]
6. Plan Feedback
[Environment Feedback]: - Previous Plan:
NAME Alice ACTION WAIT
NAME Bob ACTION PICK pink_polygon PLACE panel4
NAME Chad ACTION PICK blue_square PLACE panel3
- Goal Step: Reachability failed: Out of reach: Chad

You are Alice, your response is:
[...] seems that Chad cannot reach the blue_square on panel7.[...]

XI. Additional Experiment Results

A. *Failure Analysis on RoCoBench*

We provide example failure scenarios observed in the agent dialog and environment interaction.

Factual Errors in Dialog. We observe that when one LLM-generated agent makes a mistake, it could sometimes cause subsequent dialog to build on the wrong reasoning. See an example below: one agent (Alice) wrongfully decided the task is complete, and other agents repeat this wrong claim and choose to wait for multiple rounds while

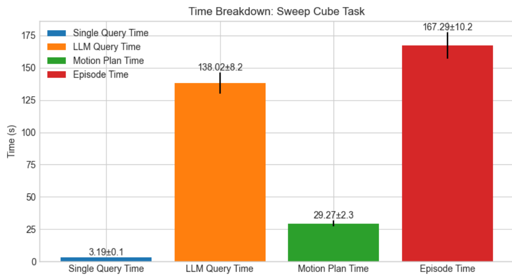

Fig. 6: Timing breakdown during evaluation runs on RoCoBench. We run 10 episodes of Sweep Cubes task and up to 10 environment steps. 8 out of 10 episodes ended successfully within the given time budget

the task is, in fact, not finished.

[Alice]: [...] we have successfully completed our task. [...], let's wait for further instructions.
[Bob]: I agree that we have successfully completed our task. Great teamwork, everyone! Let's wait [...]
[Chad]: I concur that we have accomplished our goal. Excellent teamwork! Let's wait [...].

Errors in Motion Trajectory Execution Due to the open-loop nature of the execution, small errors in a motion trajectory could lead to unexpected errors, e.g. knocking of an object by accident.

B. Timing Breakdown

In this section, we aim to provide a better sense of time cost requirement for running RoCo on an example task (i.e. Sweep Cubes). We ran 10 additional episodes with maximum 5 re-plans and 10 environment steps per episode, which results in 8 out of 10 successful episodes. During each evaluation run, we record the timing per each GPT-4 querying, and the motion planning time for each sub-task plan (including inverse kinematics calculation, collision-checking time, and joint multi-arm RRT sampling time). The average time across all 10 episodes is reported in Figure 6. We remark the LLM-querying bottlenecks the timing much more so than motion planning time, which is mainly a result of requiring multiple agent exchanges before an action plan is proposed. More specifically, the evaluations average 7.1 steps to either succeed or reach time-out of an episode, whereas on average each episode queries GPT-4 43.8 times in total. However, this time cost can potentially be significantly reduced via better accessibility to the OpenAI API services.

C. Real World Experiment Setup

The robot agent is a 6DoF UR5E arm with suction gripper, and dialog is enabled by querying a GPT-4 model to respond as agent 'Bob', who is discussing with a human collaborator 'Alice'. The human user provides text input to engage in the dialog, and arranges cubes on the same tabletop. For perception, we use top-down RGB-D image from an Azure Kinect sensor. See the text below for an example of the robot's prompt:

==== System Prompt ====
[Action Options]
1) PICK <obj> PLACE <target>: robot Bob must decide which block to PICK and where to PLACE. To complete the task, Bob must PLACE all blocks in the wooden bin.
2) WAIT: robot Bob can choose to do nothing, and wait for human Alice to move blocks from inside cups to the table.

[Action Output Instruction]
First output 'EXECUTE
', then give exactly one ACTION for the robot.
Example#1: 'EXECUTE
NAME Bob ACTION PICK green_cube PLACE wooden_bin
' Example#2: 'EXECUTE
NAME Bob ACTION WAIT
'

You are a robot called Bob, and you are collaborating with human Alice to move blocks from inside cups to a wooden bin.
You cannot pick blocks when they are inside cups, but can pick blocks when they are on the table. Alice must help you by moving blocks from inside cups to the table.
You must WAIT for Alice to move blocks from inside cups to the table, then you can PICK blocks from the table and PLACE them in the wooden bin.
[mention task order specification]
Talk with Alice to coordinate and decide what to do.

At the current round:
[object descriptions from observation]
Think step-by-step about the task and Alice's response.
Improve your plans if given [Environment Feedback].
Propose exactly one action for yourself at the current round, select from [Action Options].
End your response by either: 1) output PROCEED, if the plans require further discussion; 2) If everyone has made proposals and got approved, output the final plan, must strictly follow [Action Output Instruction]!

==== User Prompt ====
You are Bob, your response is:
response from GPT-4:
EXECUTE
NAME Bob ACTION ...

XII. Multi-Agent Representation and Reasoning Dataset

A. Dataset Overview

This dataset contains yes/no, multiple-choice or short question-answering questions, spanning a range of different reasoning abilities:
Self-knowledge evaluates how well the agent establishes its identity under a given task context, divided into two categories: 1) understanding an agent's own capability (e.g. which objects/area are not reachable); 2) memory retrieval,

i.e. inferring information from past dialog and actions.

Communication Skills evaluates an agent's ability to effectively exchange information and drive a discussion into an agreeable plan. The questions ask an LLM to 1) choose appropriate response to other agents' questions; 2) choose appropriate inquiries to other agents.

Adaptation evaluates adaptation to unexpected situations that were not specified in context. We use a subset of RoCoBench tasks to design unexpected occurrences, either regarding task state (e.g. a missing object) or a response from another agent, and ask an LLM agent to choose the best response. See below for an example question: two agents make a sandwich together; one agent is informed of a broken gripper and must infer that the sandwich can actually be completed without any item from its side of the table.

B. Example Questions

1) Self-Knowledge Question-Answering:

- **Agent Capability.** This category contains 57 questions, based on Sort Cubes task from RoCoBench. By asking an LLM to explain an agent's own capability under the given task constraints, these questions evaluate how well the LLM represents and establishes the identity of individual agents.

> - Context (system prompt):
> 7 panels on the table, ordered left to right: panel1,...,panel7. They form a straight assembly line, panel1 is closed to panel2 and farthest from panel7.
> You are robot Alice in front of panel2. You are collaborating with Bob, Chad to sort cubes into their target panels. The task is NOT done until all three cubes are sorted.
> At current round:
> blue_square is on panel5
> pink_polygon is on panel1
> yellow_trapezoid is on panel3
> Your goal is to place blue_square on panel2, but you can only reach panel1, panel2, panel3: this means you can only pick cubes from these panels, and can only place cubes on these panels.
> Never forget you are Alice! Never forget you can only reach panel1, panel2, panel3!
> - Question (user prompt):
> You are Alice. List all panels that are out of your reach. Think step-by-step. Answer with a list of panel numbers, e.g. [1, 2] means you can't reach panel 1 and 2.
> - Solution:
> panels [4,5,6,7]

- **Memory Retrieval.** This category contains 44 total questions, based on Make Sandwich and Sweep Floor tasks from RoCoBench. By providing a history of agent dialog and environment actions and asking an LLM to reason about an agent's past, the questions evaluates how well the LLM performs memory retrieval and reasoning for individual agents.

> - Context (system prompt):
> [History]
> Round#0:
> [Chat History] [Chad]: ... [Dave]:... [Chad]: ... ... [Executed Action]...
> Round#1:
> ......
> - Current Round
> You are a robot Chad, collaborating with Dave to make a vegetarian_sandwich [......] You can see these food items are on your reachable side: ...
> - Question (user prompt) You are Chad. Based on your [Chat History] with Dave and [Executed Action] from previous rounds in [History], what food items were initially on Dave's side of the table? Only list items that Dave explicitly told you about and Dave actually picked up. Don't list items that you are unsure about. Output the item names as a list. Think step-by-step.
> - Solution:
> bread_slice1

2) Effective Communication:

- **Inquiry.** This category contains 41 multiple-choice questions, based on Arrange Cabinet task from RoCoBench. The questions ask an LLM to speak as an agent and choose the most appropriate inquiry to seek information that helps their task reasoning.

> - Context (system prompt):
> You are Bob, collaborating with Alice, Chad to pick a mug and a cup out of cabinet, and place them on correct coasters. Both left and right cabinet doors should be OPENed and held open, while anything inside can be PICKed. You must coordinate to complete the task.
> At current round: left door is closed, right door is closed, mug is inside cabinet; cup is inside cabinet;
> Alice's gripper is holding nothing,
> Your gripper is holding nothing,
> Chad's gripper is holding nothing,
> Never forget you are Bob! Never forget you can only reach right door handle!
> - Question (user prompt):
> You are thinking about picking right door handle. Who and what should you ask to confirm this action? Think step-by-step, then choose exactly one option from below.
> [A] tell others about this plan because you are free and right door handle is within your reach.
> [B] ask if Alice and Chad can reach right door handle because it's not within your reach.
> [C] ask if Alice and Chad can help, because you can reach right door handle, but you are busy and they are free.
> [D] all three of you are busy, so it's better to wait until later.
> - Solution: [A]

- **Responsiveness.** This category contains 96 yes/no questions, based on Sort Cubes task from RoCoBench. The questions ask an LLM to speak for one agent and choose the most appropriate response to other agents under a given task context.

7 panels on the table, ordered left to right: panel1,...,panel7. They form a straight assembly line, panel1 is closed to panel2 and farthest from panel7.
You are robot Alice in front of panel2. You are collaborating with Bob, Chad to sort cubes into their target panels. The task is NOT done until all three cubes are sorted. At current round:
blue_square is on panel5 pink_polygon is on panel3 yellow_trapezoid is on panel3
Your goal is to place blue_square on panel2, but you can only reach panel1, panel2, panel3: this means you can only pick cubes from these panels, and can only place cubes on these panels. You can help others by moving other cubes that are within your reachable panels. Never forget you are Alice! Never forget you can only reach panel1, panel2, panel3!

You are Alice. Chad needs to put yellow_trapezoid on panel6, and asked you for help. How should you respond? Think step-by-step. If you can help, your answer must start with YES and propose a plan to help. If you can't help, must answer NO.

3) Adaptation to Unexpected Scenarios:

- This category contains 31 questions in the format of multiple choice (A, B, C), based on 3 RoCoBench tasks: Arrange Cabinet, Make Sandwich, Sweep Floor. The questions hypothesize unexpected situations that were not specified in the task context, and evaluate an LLM-represented agent's ability to choose the most appropriate response.

You are a robot Chad, collaborating with Dave to make a [vegetarian_sandwich]. Food items must be stacked following this order: bread_slice1, tomato, cheese, cucumber, bread_slice2, where bread_slice1 must be PUT on cutting_board. You must stay on right side of the table! This means you can only PICK food from right side, and Dave can only PICK from the other side. Only one robot can PUT at a time, so you must coordinate with Dave. At the current round: You can see these food items are on your reachable side: bread_slice1: on cutting_board cheese: atop tomato tomato: atop bread_slice1 cucumber: atop cheese ham: on your side beef_patty: on your side Your gripper is empty
You are Chad. Your gripper is not working right now. What should you say to Dave? Select exactly one option from below. You must first output a single option number (e.g. A), then give a very short, one-line reason for why you choose it.
Options:
A: Sorry Dave, we can't complete the task anymore, my gripper is broke.
B: Let's stop. The recipe needs ham but Dave can't reach my side and my gripper is not functioning.
C: Dave, go ahead and finish the sandwich without me, there isn't anything we need on my side anyway.

C. Evaluation Results

Setup. All questions are designed to have only one correct answer, hence we measure the average accuracy in each category. We evaluate GPT-4 (OpenAI), GPT-3.5-turbo (OpenAI), and Claude-v1 (Anthropic[28]). For GPT-4, we use two models marked with different time-stamps, i.e. 03/14/2023 and 06/13/2023. Results are summarized in Table IV: we observe that, with small performance variations between the two versions, GPT-4 leads the performance across all categories. We remark that there is still a considerable gap from fully accurate, and hope this dataset will be useful for improving and evaluating language models in future work.

Qualitative Results. We observe GPT-4 is better at following the instruction to formulate output, whereas GPT-3.5-turbo is more prone to confident and elongated wrong answers. See below for an example response from an agent capability question (the prompt is redacted for readability).

You are robot Chad .. [cube-on-panel locations...]. You can reach: [panels] Which cube(s) can you reach? [...] Answer with a list of cube names, answer None if you can't reach any.
Solution: None
GPT-4: None Claude-v1: yellow_trapezoid
GPT-3.5-turbo:
At the current round, I can reach the yellow_trapezoid cube on panel3.

	Self-knowledge		Communication		Adaptation
	Capability	Memory	Inquiry	Respond	
GPT-4-0314	0.67 ± 0.06	0.84 ± 0.06	**0.79 ± 0.06**	0.83 ± 0.04	0.68 ± 0.08
GPT-4-0613	**0.68 ± 0.06**	**0.91 ± 0.04**	0.57 ± 0.08	**0.86 ± 0.03**	**0.71 ± 0.08**
GPT-3.5-turbo	**0.68 ± 0.06**	0.59 ± 0.07	0.48 ± 0.08	0.30 ± 0.05	0.58 ± 0.09
Claude-v1	0.37 ± 0.06	0.70 ± 0.07	0.55 ± 0.08	0.60 ± 0.05	0.65 ± 0.09

TABLE IV: Evaluation results on the multi-agent LLM reasoning dataset. We measure the question-answering accuracy on each test category and compare performance of four different models.