

# Mixed-Initiative Dialog for Human-Robot Collaboration

Anonymous Author(s)

Affiliation

Address

email

**Abstract:** Effective robotic systems for long-horizon human-robot collaboration must adapt to a wide range of human partners, whose physical behavior, willingness to assist, and understanding of the robot’s capabilities may change over time. This demands a tightly coupled communication loop that grants both agents the flexibility to propose, accept, or decline requests as they coordinate toward completing the task effectively. We apply a **Mixed-Initiative** dialog paradigm to **Collaborative human-roBot** teaming and propose **MICoBot**, a system that handles the common scenario where both agents, using natural language, take initiative in formulating, accepting, or rejecting proposals on who can best complete different steps of a task. To handle diverse, task-directed dialog, and find successful collaborative strategies that minimize human effort, MICoBot makes decisions at three levels: (1) a meta-planner considers human dialog to formulate and code a high-level collaboration strategy, (2) a planner optimally allocates the remaining steps to either agent based on the robot’s capabilities (measured by a simulation-pretrained affordance model) and the human’s estimated availability to help, and (3) an action executor decides the low-level actions to perform or words to say to the human. Our extensive evaluations in simulation and real-world—on a physical robot with 18 unique human participants over 27 hours—demonstrate the ability of our method to effectively collaborate with diverse human users, yielding significantly improved task success and user experience than a pure LLM baseline and other agent allocation models. More information on our website: <https://mico-bot.github.io/>

**Keywords:** Human-Robot Interaction, Mixed-Initiative Grounded Dialog



Figure 1: We present MICoBot, a system for human-robot collaborative settings where both agents can initiate and carry out physical and verbal actions to negotiate how to accomplish a task together. Our system supports *both* robot-initiated (top row) *and* human-initiated (bottom row) task-directed dialog, where both agents discuss who is best suited to perform steps in a long-horizon task. The trace shows real dialog and physical interactions from our user studies (see our video and website).

## 24 1 Introduction

25 We aim to build robots that can seamlessly collaborate with humans in everyday household tasks.  
26 Such collaboration is critical for deploying today’s robots with limited capabilities. To be a truly  
27 effective partner, a collaborative robot must strive for task success with minimal human effort, while  
28 adapting dynamically to a human user’s capabilities, preferences, and willingness to help. Beyond  
29 adapting, the robot must also learn to communicate the real-time delegation of task components  
30 based on which agent—human or robot—is better suited for each. Existing approaches fall short  
31 of this ideal: modern AI assistants respond only to human-initiated interactions [1, 2], while prior  
32 human-robot interaction (HRI) solutions often assume full control over the collaboration plan and  
33 complete willingness from the human partner [3]. We argue that a truly collaborative human-robot  
34 team requires a paradigm shift towards a model where both agents can take initiative to propose, bar-  
35 gain, and accept or reject proposals from each other as they discuss in natural language how to best  
36 complete a task. In this paper, we introduce such a collaborative system, MICoBot (Mixed-Initiative  
37 Collaborative roBot), which we believe is the first to enable mixed-initiative natural language dialog  
38 for real-world physical collaboration between robots and humans.

39 MICoBot enables mixed-initiative dialog to negotiate the allocation of task steps between a human  
40 and a robot collaborating on a physical task, and to coordinate the physical and verbal actions needed  
41 to execute the plan. We formulate this task-allocation problem as a constrained optimization where  
42 the goal is to find the most suitable agent to perform each step of the task, maximizing success while  
43 minimizing human effort, as well as respecting the human-initiated requests. To handle a wide  
44 range of dialog, MICoBot makes optimization decisions across three levels. First, a meta-planner  
45 determines the high-level strategy for collaborating with the human, incorporating human-imposed  
46 constraints (such as steps they want themselves or the robot to perform), and creating reactive code  
47 to generate the necessary robot actions (verbal or physical). This layer is implemented as an LLM-  
48 based coder that generates adaptive planning code. Second, a planner executes the generated code  
49 to determine the optimal collaboration approach, taking into account the current environment state,  
50 a self-assessment of the robot’s capabilities via an affordance model trained in simulation, and a  
51 dynamic evaluation of the human’s availability and willingness based on prior interactions. Finally,  
52 an action executor carries out the next step of the plan, which could involve either performing a  
53 manipulation action or initiating/responding to a dialog with the human.

54 Through extensive experimental evaluation, we thoroughly validate our system in both simulation  
55 (with LLM-simulated humans of varying willingness and responsive moods) and the real world  
56 through a user study involving 18 unique participants collaborating with a Tiago mobile manipulator  
57 on three household tasks. Our approach outperforms a pure LLM baseline by **60%** in success rate,  
58 while reducing the amount of human effort required compared to an all-human oracle by **60%**.  
59 Additionally, our method was preferred over the pure LLM baseline by at least **80%** of participants.

60 In summary, our work’s contributions are four-fold: (1) we introduce a **new problem setting** that in-  
61 tegrates mixed-initiative natural language dialog with mixed-initiative human-robot interaction; (2)  
62 we propose a novel **optimization framework** for task allocation, balancing human and robot effort  
63 and success through a unified metric; (3) we provide a **new simulator** for collaborative household  
64 tasks built on top of MiniBehavior [4] that includes LLM-controlled virtual humans and is available  
65 on [our website](#); and (4) **robotic system and framework**: we develop MICoBot, a three-level hier-  
66 archical solution for mixed-initiative speech2speech human-robot collaboration that flexibly adapts  
67 to a wide range of real human collaborators in physically grounded, long-horizon tasks.

## 68 2 Related Work

69 **Mixed-initiative dialog** [5–7] refers to communication with freeflowing questions and answers from  
70 both parties. In the NLP field, the dominant chatbot paradigm adopted by large language models  
71 (LLMs) largely eschews mixed-initiative interaction: humans pose substantive questions, and the  
72 chatbot primarily responds to fulfill these requests [1, 2]. Recent work has sought to make dialog

73 systems more goal-directed and proactive by incorporating mixed-initiative strategies—for example,  
 74 persuading users to donate to charity, enhancing users’ emotional well-being [8–11], or clarifying  
 75 ambiguous human requests [12–14]. However, none of these systems addressed mixed-initiative  
 76 dialog in grounded, real-world collaborative scenarios involving physical manipulation tasks.

77 In the human-robot interaction (HRI) field, researchers have developed **human-robot collaboration**  
 78 **systems** that interact through language but are restricted to **single-initiative dialog**. Some of these  
 79 systems integrate LLMs as task planners or delegators [15–17] for tasks like real-world cooking [15]  
 80 and object sorting [16]. Other systems implement a leader-follower paradigm in simulated worlds,  
 81 where the leader issues natural language instructions for the follower to execute [18–21]. Single-  
 82 initiative HRI systems can ask humans for clarification [22] or assistance [23–25], or inform humans  
 83 of their observations [26–28]. However, by supporting only single-initiative dialog, these systems  
 84 lack the capacity to adapt to the evolving nature of the human, robot, and environment—limiting  
 85 their capacity to find the optimal division of labor that respects user preferences. [16].

86 Some works in HRI have explored **mixed-initiative collaborative systems without dialog**, only  
 87 with physical actions [29–34]. In particular, Baraglia et al. [35] studied separate regimes of agent  
 88 initiative (human-initiative, requesting help, or robot-initiative, proactively helping), but failed to  
 89 support a natural human-robot dialog. By focusing solely on physical actions, these prior works  
 90 overlook the critical role of communication in effective collaboration, thereby limiting the flexibility  
 91 of the human-robot team. With MICoBot, we enable both agents to take initiative—through both  
 92 physical and verbal actions—via task-grounded dialog.

93 Several prior works in robotics and planning have studied the problem of **human-robot optimal**  
 94 **task allocation**, typically optimizing the time to perform a task or minimizing idle agents, posing  
 95 the problem as a scheduling problem [36, 37]. Others have prioritized different objectives, such as  
 96 safety [38], through the formulation of a constrained optimization problem [39]. While these solu-  
 97 tions may achieve shorter execution times, they assume a priori known capabilities and availability  
 98 of all agents, including both robots and humans. In contrast, MICoBot can adapt to the specific  
 99 human’s willingness to help by estimating its availability based on previous dialog.

### 100 3 Problem Setting: Task Collaboration with Mixed-Initiative Dialog

101 **MDP Formulation.** In this paper, we study human-robot  
 102 collaboration for shared manipulation problems where  
 103 mixed-initiative dialog occurs. In these problems, we as-  
 104 sume that both agents can observe the state of the world,  
 105  $s \in \mathcal{S}$ , and perform actions,  $a \in \mathcal{A} = \mathcal{A}_p \cup \mathcal{A}_v$ , com-  
 106 prised of a physical action space,  $\mathcal{A}_p$ , that directly af-  
 107 fects the physical state of the environment  $s$ , (e.g., move  
 108 objects, open them, etc.), and a free-form, natural lan-  
 109 guage verbal action space,  $\mathcal{A}_v$ , that are directly observed  
 110 by the other agent but do not change the physical state.

111 We model the problem as a Markov Decision Process (MDP) from the robot’s point of view (see  
 112 Fig. 2), where on each environment step, the robot performs some action,  $a_R \in \mathcal{A}_{p,R} \cup \mathcal{A}_{v,R}$  and  
 113 receives an observation  $o = [I, a_{v,H}, s_{proprio}]$  consisting of an RGB-D image  $I$ , an optional verbal  
 114 action from the human partner  $a_{v,H}$ , and the robot’s proprioceptive state  $s_{proprio}$ . Within each en-  
 115 vironment step, the human may perform a series of actions,  $a_H \in \mathcal{A}_{p,H} \cup \mathcal{A}_{v,H}$ , in its own physical  
 116 and verbal action space after perceiving the world state and robot’s previous dialog,  $a_{v,R}$ .

117 **Physical and Verbal Action Spaces.** The physical and verbal action spaces,  $\mathcal{A}_p$  and  $\mathcal{A}_v$ , are shared  
 118 between both agents. Each element of these action space are a parameterized action primitive rep-  
 119 resented by the pair,  $a_{p/v} = (\omega_{p/v}, \theta_{p/v})$ .  $\omega_p$  is the type of the physical action primitive (open,  
 120 pick-and-place, etc.) and  $\theta_p$  are the corresponding parameters (e.g., what object to open or pick  
 121 and where to place it). We assume that humans are fully competent in executing all steps of a col-  
 122 laborative household manipulation task, but may be unwilling or unavailable to perform some or

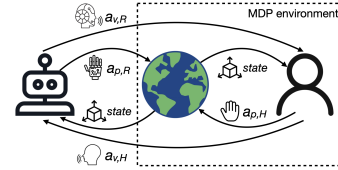


Figure 2: Our MDP Formulation for Mixed-Initiative Collaboration

all required actions. Their behavior can range from indifferent (never acting) to overly proactive (completing the entire task without robot involvement). In contrast, robots often have limited manipulation capabilities and may be unable to execute more complex actions.  $\omega_v$  is the type of the verbal action primitive (`ask_human_for_help`, `respond_to_human`, etc.) and  $\theta_v$  are the corresponding parameters that define the necessary context of the verbal primitive (e.g., what step the robot needs help on, or can/cannot perform). While the types of verbal actions are limited, the generated language based on them is freeform and open-vocabulary. MICOBot first selects an abstract verbal action from this space, then translates it into a natural language utterance to negotiate with the human—conveying its requests and the assistance it requires for successful collaboration. MICOBot must reason over the asymmetries in physical capabilities to devise collaboration strategies and negotiate them with verbal actions, maximizing task success while minimizing human effort.

**Collaborative Task Definition and Problem Statement.** We assume the collaborative task is defined by a task plan of length  $K$ , known to both agents and represented as a sequence of unsigned **physical** action primitives,  $[a_{p,0}, \dots, a_{p,K-1}]$ , such as `[(pick-and-place(box, table), \dots, close(box))]`, obtained from the task instructions or off-the-shelf task planner. To complete the manipulation task while minimizing human effort, the system must allocate steps of the plan between the two agents—negotiating with the human through robot-initiated dialog to suggest assignments, adapting to human preferences through human-initiated dialog, and ultimately executing its assigned physical actions. At each step  $t$ , the system must compute the best allocation of the remaining steps of the plan,  $G = [g_t, \dots, g_{K-1}]$ , where  $\forall t, g_t \in \{H, R\}$ . The optimal allocation  $G^*$  maximizes the expected task success probability while minimizing total human effort. These objectives are inherently competing: a policy focused solely on maximizing success might allocate all steps to the human (assumed to be perfectly competent); conversely, minimizing human effort alone would assign all steps to the robot, even when it may be incapable of completing certain steps. The optimization also incorporates constraints conveyed through the mixed-initiative dialog history, such as task allocation requests or proposed task splits. The resulting allocation  $G^*$  determines whether the robot executes the current step ( $R$ ) or negotiates with the human for assistance ( $H$ ).

## 4 MICOBot: Mixed-Initiative Collaborative Robot

### Collaborative Task Allocation as Constrained Optimization.

In MICOBot, we formulate the step allocation problem for collaborative tasks as a constrained optimization, where the objective is to maximize expected task success while minimizing human effort. Constraints—such as preferences for certain steps to be done by a particular agent—are inferred through dialog with the human. To simplify the optimization and avoid a complex multi-objective formulation, we combine success probability and effort into a single cost metric for each step, regardless of whether it is performed by the robot or the human. Building on prior work on temporal distances in reinforcement learning [40], we use Q-functions to unify these two components. We assume each task step is executed by a multi-task policy  $\pi$  that operates at a fixed control frequency (e.g., once per second), performing continuous low-level control. In this low-level MDP (distinct from the high-level task MDP described in Sec. 3), we define the reward as  $r = -1$  per time step until the skill completes or times out, at which point  $r_{termination} = 0$ . A well-trained Q-function,  $Q : o_t \times a_t = (\omega_t, \theta_t) \mapsto \mathbb{R}$  with a discount factor of 1, then represents the **negative expected number of timesteps** until skill completion from a given state. For a highly competent agent that never fail (e.g., a human), this corresponds to the average timesteps required to perform the action. For an imperfect agent that may fail, the Q-function reflects a weighted expectation over both successful and failed outcomes—where failure contributes a significant timestep penalty (timeout) weighted by its probability. We assign each agent a distinct Q-function:  $Q_R$  for the robot and  $Q_H$  for the human. These agent-specific Q-functions thus provide a unified, interpretable cost metric for comparing step allocations, jointly capturing both execution time (effort) and likelihood of success.

However, directly optimizing step allocation using only the Q-functions described above introduces three key limitations that diverge from realistic human-robot collaboration scenarios: (1) human

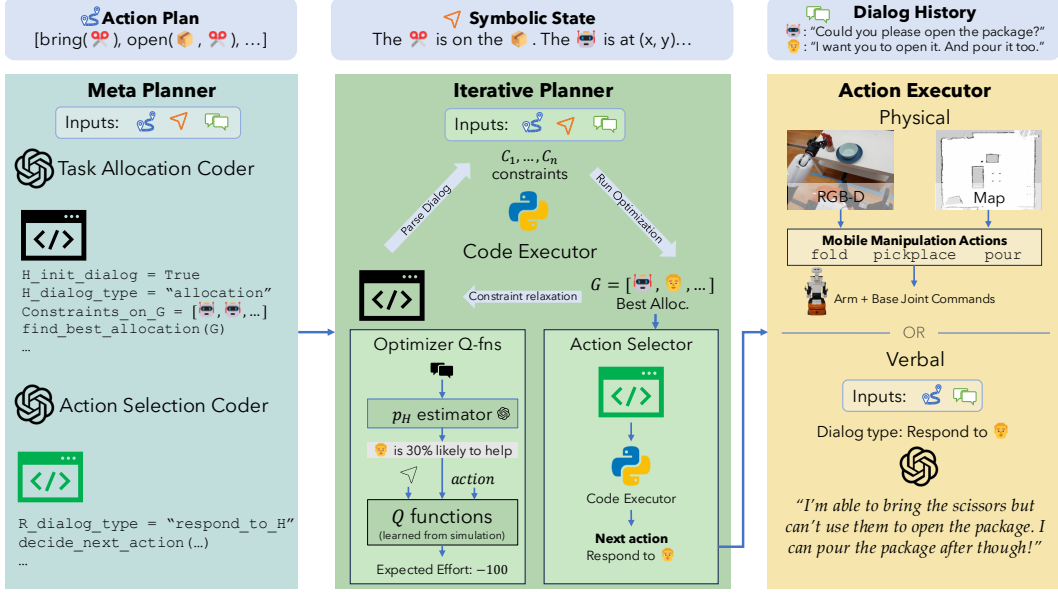


Figure 3: MICoBot consists of 3 decision-making modules: a meta-planner that outputs a strategy for task collaboration expressed through adaptive planning code, a planner that executes the code and optimizes our objective in Equation 1 to find the next primitive action to take, and the action executor that outputs the low-level physical action trajectory or verbal utterance to say to the human.

and robot effort are treated as equally, ignoring the higher value typically placed on human time and attention; (2) the human is assumed to always comply with robot-initiated requests, overlooking variability in willingness or availability; and (3) human-initiated requests or preferences are not taken into account, limiting the system’s ability to adapt to human intent. To address (1), we introduce a *human-effort factor*,  $\alpha$ , indicating how much more valuable human effort is compared to robot effort. To address (2), we adjust the human Q-values by incorporating an inferred probability  $p_{H,t}$  that represents the likelihood of the human agreeing to perform action  $a_{H,t} = \omega_t(\theta_t)$  when asked. For less cooperative users, this probability lowers the expected success of the action, effectively increasing the magnitude of the negative Q-value due to potential human refusal. To address (3), we treat the optimization problem as subject to constraints,  $C_1, \dots, C_n$ , extracted from human-initiated dialog—such as explicit requests to perform specific steps themselves or to delegate them to the robot. Altogether, we propose the following objective to find the optimal task allocation  $G$ :

$$\begin{aligned} \max_{g_t, \dots, g_T} \quad & \sum_t^{T-1} \left( \mathbb{1}_{g_t=H} \cdot \frac{\alpha}{p_{H,t}} + \mathbb{1}_{g_t=R} \right) Q_{g_t}(s_t, a_t), \\ \text{s.t.} \quad & C_1, \dots, C_n \leq 0. \end{aligned} \quad (1)$$

that minimizes expected time-to-success while prioritizing saving human effort.

#### 4.1 MICoBot Framework

MICoBot is a three-level framework (Fig. 3) that includes 1) a meta-planner, which parses previous human dialog and generates code to optimize for task allocation and select the next action for the robot, 2) an iterative planner, which updates planning state variables, allocates and decides the next action to perform, by executing the code, and 3) an action executor, which carries out the action primitive, either through low-level physical actions or with an utterance to say to the human.

**L1: Meta-planner.** The meta-planner dictates the overall strategy for the lower levels to follow. Based on the most recent human dialog, the current symbolic state of the world, the task plan, and approximately 15 in-context learning (ICL) examples, it generates two pieces of code: first, **task**



195 **allocation** code to adapt the optimization computation, such as to map human dialog into additional  
 196 constraints, and second, **action selection** code, to determine how to choose the next action, such as  
 197 whether to engage in additional dialog before making further progress on the plan. The meta-planner  
 198 is implemented as an LLM-based (GPT-4o) coder.

199 **L2: Iterative Planner.** The iterative planner executes code generated by the meta-planner in two  
 200 stages. *In the first stage*, it runs the optimization routine, which enumerates all possible task al-  
 201 locations and selects the one that maximizes the objective in Eq. 1. To compute this, the planner  
 202 instantiates agent-specific Q-functions based on the current state and candidate actions (see below  
 203 for details on Q-function), and estimates the probability of human assistance,  $p_{H,t}$ , using an LLM-  
 204 based sentiment analysis over the prior human-robot dialog. By adjusting  $p_{H,t}$ , MICoBot adapts  
 205 to varied user sentiments by estimating the expected cost of assigning tasks to them. In the initial  
 206 iteration, the planner incorporates all constraints produced by the meta-planner from the mixed-  
 207 initiative dialog history. If no feasible allocation is found—for instance, if a human insists the robot  
 208 perform a step it cannot complete—the planner iteratively relaxes the most recent constraint from  
 209 human dialog. *In the second stage*, once the optimal allocation is determined, the planner invokes  
 210 meta-planner code to generate the optimal action—verbal or physical— $a = (\omega, \theta)$  to execute.

211 **L3: Action Executor.** The action executor is responsible for executing the action primitive selected  
 212 by the planner. For physical actions, it generates a trajectory for navigation and arm movement to  
 213 reach the location and manipulate the target object while avoiding obstacles. Following a similar  
 214 pipeline to Shah et al. [41], we use the move\_base ROS package for path planning over a 2D  
 215 occupancy map, and Grounding DINO [42] to segment the target object from the scene based on the  
 216 natural language query in  $\theta_t$ . An RGB-D camera is used to backproject segmented image pixels into  
 217 a 3D point cloud, from which we identify graspable or placeable points in the robot’s workspace.  
 218 Inverse kinematics (IK) is then used to move the arm to these points. For verbal actions, we employ  
 219 GPT-4o to generate natural language utterances to communicate with the human, based on both the  
 220 intended dialog intent (e.g., help request, split proposal) and the verbal action parameters (context  
 221 required for appropriate generation) from the upstream planner. Using approximately 10 in-context  
 222 learning (ICL) examples, the LLM produces free-form language grounded in the task context.

223 **Training Q-functions.** MICoBot’s optimization process depends on accurate approximations of  
 224 the Q-functions, which capture each agent’s expected effort and likelihood of success when execut-  
 225 ing a task step. To estimate the robot’s Q-function ( $Q_R$ ), we use the OmniGibson simulator [43],  
 226 configured with a coarse model of the real-world task and environment (see Appendix for visual-  
 227 izations). In simulation, we execute action primitives that closely mirror those used by the physical  
 228 robot, recording both completion times and failure cases. These statistics are used to construct  $Q_R$   
 229 as described earlier in this section. Since the task state is represented symbolically in both the simu-  
 230 lator and the real-world system, the sim-to-real gap is minimal. To estimate the human’s Q-function  
 231 ( $Q_H$ ), we assume humans do not fail at task execution. Thus, we only require time estimates for  
 232 each step, which MICoBot obtains by prompting an LLM to predict how long a human would take  
 233 to execute action  $a_t = \omega_t(\theta_t)$ , plus a travel time estimate based on the human-object distances.

234 **Hierarchical Plan.** To improve communication for long-horizon task plans, MICoBot groups adja-  
 235 cent low-level steps into semantically meaningful abstract actions that can be discussed more suc-  
 236 cinctly with the human. The system only descends to a finer-grained level of detail when neces-  
 237 sary—during negotiation over low-level step assignments. This hierarchical approach reduces the  
 238 frequency and complexity of dialog, resulting in more efficient and user-friendly communication.

## 239 5 Evaluation

240 We evaluate MICoBot in both real-world and simulated settings. In the real world, a Tiago mobile  
 241 manipulator collaborates with a human user on household manipulation tasks. In simulation, we use  
 242 the Mini-Behavior gridworld [4] with a simulated human, allowing for larger-scale experimentation  
 243 and controlled comparisons across methods, particularly in relation to human behavior and dialog  
 244 dynamics. As suggested before, a successful robotic collaborator must complete the task efficiently

	Pour Package in Bowl $n = 6$		Assemble Toy Car $n = 6$		Pack Gift Box $n = 6$		Average $n = 18$	
	Ours	LLM	Ours	LLM	Ours	LLM	Ours	LLM
Entire Task Success Rate (% , $\uparrow$ )	<b>50</b>	0	<b>67</b>	0	<b>67</b>	0	<b>61.1</b>	0.0
% of task steps completed ( $\uparrow$ )	<b>83</b>	60	<b>94</b>	29	<b>88</b>	50	<b>88.2</b>	46.4
% of steps performed by Human	21	5	60	5	35	21	38.8	10.4
% Users Preferring ... ( $\uparrow$ )	<b>83</b>	17	<b>100</b>	0	<b>67</b>	33	<b>83.3</b>	16.7
Communicative ability ( $\uparrow$ )	<b>3.3/5</b>	2.3/5	<b>4.3/5</b>	1.3/5	<b>2.8/5</b>	2.3/5	<b>3.5/5</b>	2.0/5
“Clearly communicated to me when it couldn’t do something.” ( $\uparrow$ )	<b>4.3/5</b>	2.3/5	<b>3.7/5</b>	1.2/5	<b>4.2/5</b>	2.5/5	<b>4.1/5</b>	2.0/5
Overall Satisfaction working w/ Robot ( $\uparrow$ )	<b>3.7/5</b>	2.7/5	<b>3.5/5</b>	1.5/5	<b>3.5/5</b>	2.5/5	<b>3.6/5</b>	2.2/5

Table 1: Comparison between our method and the LLM baseline across three real-world tasks. Ratings out of 5 are on the Likert scale. By more effective task allocation and communication, our method is able to achieve much higher task success rate and overall user satisfaction.

while minimizing human effort. Accordingly, our primary evaluation metric is the **success rate per unit of human effort**. We also report **subjective measures of robot behavior**, including user satisfaction, preference rankings, and Likert-scale ratings.

**Environment.** In the real-world, we perform our experiments in a mock apartment with a kitchen and living room area with commonplace furniture. In all of our tasks, the robot and human work together on opposite sides of a coffee table, and the human spends most of their time on the couch, where they can do their own work. The human is allowed to be as inactive or proactive as they wish and to perform physical and verbal actions as defined in Section 3 (though we continue running the trial if they initiate dialog beyond the scope). Each human user study consisted of two 20-30 minute trials, in which they collaborated with both our method and a pure LLM baseline. The ordering of the two trials was randomly determined. Trials for all methods **terminate** under any of the following conditions: an irrecoverable primitive failure occurs,  $4T$  steps have elapsed for a plan of length  $T$ , an infeasible step is allocated to the robot twice consecutively, or the human refuses twice to perform a step that the robot is incapable of executing.

**Baselines.** Because multiple components of our method are powered by LLMs, we compare our approach to a pure LLM baseline (**LLM**) given the same information as our meta-planner: symbolic state, dialog history, task plan, and  $\alpha$  human-robot effort tradeoff factor. The LLM baseline is also provided with a list of the robot’s available skills and assumes that the human always successfully completes a step once they agree to perform it. The LLM baseline is prompted to produce a plan allocation  $G$  that primarily optimizes for task success and secondarily minimizes human effort.

To control for the amount of human effort elicited in the user studies with our method, we compute an additional random allocation baseline that does not involve a human participant, **RECB** (random effort-controlled baseline). We denote the percentage of steps done by the human in the user trials of our method as  $p_c$ . RECB randomly allocates the current step to the human with probability  $p_c$ , and assumes the human always accepts the robot’s request. RECB also assumes access to oracle robot primitives with 100% success rate.

In simulation, we additionally compare against an **RL** baseline (hierarchical task allocator + robot policy; see Appendix for details), and a naive **Random** baseline, which randomly allocates either agent (with probability 50%) to perform the next step.

**Ablations.** To measure the importance of mixed-initiative, we perform the following ablations in simulation: **H-init** and **R-init**, where the human or the robot alone can initiate any dialog, respectively. We further ablate components of MICoBot in simulation by running it **w/o P<sub>H</sub>** (no  $p_{H,t}$  estimation) and **w/o Plan Hierarchy** (where our method talks to the human in low-level steps).

**Tasks.** We performed user studies on 3 real-world tasks (Pour Package into Bowl, Assemble Toy Car, and Pack Gift Box) with 6 participants per task for a total of 18 unique human participants. Each task is a long horizon sequence of 5 to 8 mobile manipulation steps. See Appendix for details.

**Experimental analysis.** Our experiments are designed to answer the following research questions:

**(1) Does our method achieve the best trade-off between task success and minimizing human effort?** In our real-world user study (Table 1), MICoBot achieves a 61% task success rate, compared

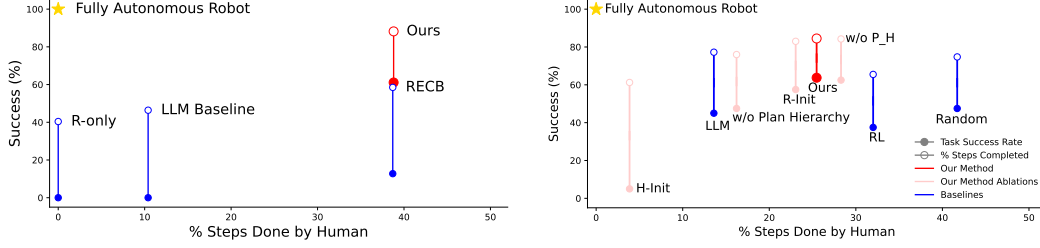


Figure 4: In both **real-world** user studies (left) and **simulation trials** with a simulated human (right), our method (red) demonstrates the best tradeoff in achieving task success (y-axis) for a given amount of human effort (x-axis) than baselines (blue) and our method’s ablations (pink).

to 0% for the LLM baseline, by leveraging human assistance on 38% of the steps. The LLM baseline underperformed because it prioritized minimizing human effort over task completion—requesting and receiving help in only 10% of steps, even when the robot lacked the capability to execute them. To control for the amount of human effort received, we compare our method to RECB in Figure 4. Despite RECB assuming oracle robot primitives with 100% success, our method still significantly outperforms it, demonstrating a more effective balance between success and human workload.

## (2) How do users feel about working with our system?

The A/B blind preference test in Table 1 shows that 83% of users preferred our method over the LLM baseline. Our method also significantly outperformed the baseline in user scores on overall satisfaction, communicative ability, and capability in asking for a suitable amount of help (See Figure 5). In contrast, the LLM baseline consistently failed to express when it needed help and was often unwilling to reject tasks it could not complete, leading to over-promises and task failures. A representative dialog exchange—available in the Appendix and on our project website—shows MICoBot successfully persuading an initially reluctant user to perform a step the robot was incapable of executing.

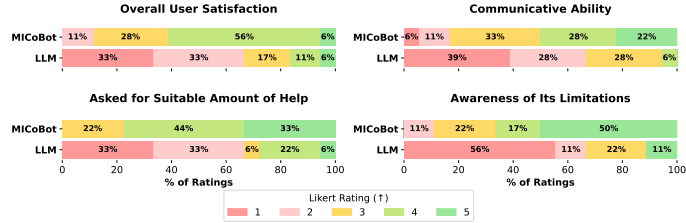


Figure 5: Our method substantially outperforms the pure LLM baseline in user ratings averaged over all  $n = 18$  participants. In contrast, the LLM baseline consistently failed to express when it needed help and was often unwilling to reject tasks it could not complete, leading to over-promises and task failures. A representative dialog exchange—available in the Appendix and on our project website—shows MICoBot successfully persuading an initially reluctant user to perform a step the robot was incapable of executing.

**(3) Is mixed-initiative dialog critical to our method’s performance?** Figure 4 (right) shows that our full method outperforms both ablated variants that restrict dialog to single-initiative modes: robot-only initiation (R-init) and human-only initiation (H-init). H-init performs especially poorly, as it prevents the robot from requesting help for steps it cannot execute. R-init performs slightly worse than the full method because it does not allow the human to proactively initiate dialog and assist when appropriate. These results underscore the importance of mixed-initiative dialog in enabling flexible, robust human-robot collaboration.

Additional experimental results and analysis (e.g. the role of  $p_{H,t}$  estimation), are in the Appendix.

## 6 Conclusion

We proposed MICoBot, a real-world robotic collaborator that can engage in mixed-initiative dialog with humans on long-horizon mobile manipulation tasks. Our work represents the first effort to unify two previously unconnected lines of research: mixed-initiative dialog and HRI. To this end, we formulated a novel optimization function and robotic framework using mixed-initiative dialog as a rich interface for task allocation to maximize task success while minimizing human effort and complying with verbally-expressed human preferences. Real-world user studies with 18 human participants and nearly a thousand trials in simulation demonstrate the efficacy, adaptability, and user satisfaction of our method across a diverse range of human physical and verbal behavior.



## 7 Limitations and Future Work

This paper represents our initial effort on uniting mixed-initiative natural-language dialog with mixed-initiative human-robot interaction. While we focused on delegating steps for long-horizon manipulation tasks in a manner that maximizes task success and minimizes human effort, we believe this paper opens up exciting new avenues for future work. These include enabling both agents learning to provide and incorporate spatial-temporal feedback to each other while performing a task, share relevant task information in an imperfect-information setting, and replan and redefine a task as necessary, all through mixed-initiative dialog interactions.

MICoBot has a number of limitations. First, it assumes a fixed plan with a predetermined ordering of steps. It cannot handle cases where the human wishes to add new steps to or remove existing steps from the plan dynamically, such as if the user tells the robot to “grab another cold drink while you’re at the fridge before coming back to me.” Our method also cannot handle cases where a robot and human wish to collaborate simultaneously on the same step in the plan, such as if the robot holds a roll of tape and the human cuts from it. Furthermore, MICoBot does not support parallelization where both the human and robot can work on different steps of a task simultaneously. One way to address this would be to operate on plan trees, where the parent nodes are steps that must be done before the child nodes, and sibling nodes can be executed by either agent in parallel.

Our method could be improved further by taking into account more information about the user. For instance, MICoBot assumes that “effort” is based on the time necessary to perform a task until completion. However, effort may also depend on the intensity of the task, how much the user enjoys it, and how physically capable each user is—our method had sidestepped this issue by assuming each human would expend the same amount of effort for each action primitive from some given state  $s$ . Finally, there are additional ways to better predict  $p_{H,t}$ , such as by processing tone-of-voice and observing facial expressions, that can enable the robot to produce more emotionally understanding dialogue, which can potentially boost task success outcomes and increase user satisfaction.

## References

- [1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [3] M. Selvaggio, M. Cagnetti, S. Nikolaidis, S. Ivaldi, and B. Siciliano. Autonomy in physical human-robot interaction: A brief survey. *IEEE Robotics and Automation Letters*, 6(4):7989–7996, 2021.
- [4] E. Jin, J. Hu, Z. Huang, R. Zhang, J. Wu, L. Fei-Fei, and R. Martín-Martín. Mini-behavior: A procedurally generated benchmark for long-horizon decision-making in embodied ai. *arXiv preprint 2310.01824*, 2023.
- [5] J. R. Carbonell. Ai in cai: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4):190–202, 1970. doi:10.1109/TMMS.1970.299942.
- [6] J. Allen, C. Guinn, and E. Horvitz. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications*, 14(5):14–23, 1999. doi:10.1109/5254.796083.
- [7] J. Chu-Carroll. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In *Sixth Applied Natural Language Processing Conference*, pages 97–104, Seattle, Washington, USA, Apr. 2000. Association for Computational Linguistics. doi:10.3115/974147.974161. URL <https://aclanthology.org/A00-1014/>.
- [8] Y. Deng, W. Lei, W. Lam, and T.-S. Chua. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv preprint arXiv:2305.02750*, 2023.
- [9] X. Yu, M. Chen, and Z. Yu. Prompt-based monte-carlo tree search for goal-oriented dialogue policy planning, 2023. URL <https://arxiv.org/abs/2305.13660>.
- [10] M. Chen, X. Yu, W. Shi, U. Awasthi, and Z. Yu. Controllable mixed-initiative dialogue generation through prompting. *arXiv preprint arXiv:2305.04147*, 2023.
- [11] Y. Deng, W. Zhang, W. Lam, S.-K. Ng, and T.-S. Chua. Plug-and-play policy planner for large language model powered dialogue agents, 2024. URL <https://arxiv.org/abs/2311.00262>.
- [12] K. Qian, A. Beirami, S. Kottur, S. Shayandeh, P. Crook, A. Geramifard, Z. Yu, and C. Sankar. Database search results disambiguation for task-oriented dialog systems, 2021. URL <https://arxiv.org/abs/2112.08351>.
- [13] Y. Deng, L. Liao, L. Chen, H. Wang, W. Lei, and T.-S. Chua. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration, 2023. URL <https://arxiv.org/abs/2305.13626>.
- [14] M. Chen, R. Sun, S. O. Arık, and T. Pfister. Learning to clarify: Multi-turn conversations with action-based contrastive self-training, 2024. URL <https://arxiv.org/abs/2406.00222>.
- [15] H. Wang, K. Kedia, J. Ren, R. Abdullah, A. Bhardwaj, A. Chao, K. Y. Chen, N. Chin, P. Dan, X. Fan, G. Gonzalez-Pumariega, A. Kompella, M. A. Pace, Y. Sharma, X. Sun, N. Sunkara, and S. Choudhury. Mosaic: A modular system for assistive and interactive cooking, 2024. URL <https://arxiv.org/abs/2402.18796>.

- [16] Z. Mandi, S. Jain, and S. Song. Roco: Dialectic multi-robot collaboration with large language models, 2023. URL <https://arxiv.org/abs/2307.04738>.
- [17] X. Feng, Z.-Y. Chen, Y. Qin, Y. Lin, X. Chen, Z. Liu, and J.-R. Wen. Large language model-based human-agent collaboration for complex task solving, 2024. URL <https://arxiv.org/abs/2402.12914>.
- [18] A. Suhr, C. Yan, C. Schluger, S. Yu, H. Khader, M. Mouallem, I. Zhang, and Y. Artzi. Executing instructions in situated collaborative interactions, 2022. URL <https://arxiv.org/abs/1910.03655>.
- [19] N. Kojima, A. Suhr, and Y. Artzi. Continual learning for grounded instruction generation by observing human following behavior, 2021. URL <https://arxiv.org/abs/2108.04812>.
- [20] D. I. A. Team, J. Abramson, A. Ahuja, A. Brussee, F. Carnevale, M. Cassin, F. Fischer, P. Georgiev, A. Goldin, M. Gupta, T. Harley, F. Hill, P. C. Humphreys, A. Hung, J. Landon, T. Lillicrap, H. Merzic, A. Muldal, A. Santoro, G. Scully, T. von Glehn, G. Wayne, N. Wong, C. Yan, and R. Zhu. Creating multimodal interactive agents with imitation and self-supervised learning, 2022. URL <https://arxiv.org/abs/2112.03763>.
- [21] Q. Gao, G. Thattai, S. Shakiah, X. Gao, S. Pansare, V. Sharma, G. Sukhatme, H. Shi, B. Yang, D. Zheng, L. Hu, K. Arumugam, S. Hu, M. Wen, D. Guthy, C. Chung, R. Khanna, O. Ipek, L. Ball, K. Bland, H. Rocker, Y. Rao, M. Johnston, R. Ghanadan, A. Mandal, D. H. Tur, and P. Natarajan. Alexa arena: A user-centric interactive platform for embodied ai, 2023. URL <https://arxiv.org/abs/2303.01586>.
- [22] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, Z. Xu, D. Sadigh, A. Zeng, and A. Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners, 2023. URL <https://arxiv.org/abs/2307.01928>.
- [23] A. Bennetot, V. Charisi, and N. Díaz-Rodríguez. Should artificial agents ask for help in human-robot collaborative problem-solving?, 2020. URL <https://arxiv.org/abs/2006.00882>.
- [24] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus. Ikeabot: An autonomous multi-robot coordinated furniture assembly system. In *2013 IEEE International Conference on Robotics and Automation*, pages 855–862, 2013. doi:10.1109/ICRA.2013.6630673.
- [25] M. Veloso, J. Biswas, B. Coltin, and S. Rosenthal. Cobots: robust symbiotic autonomous mobile service robots. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI’15, page 4423–4429. AAAI Press, 2015. ISBN 9781577357384.
- [26] D. L. Chen, J. Kim, and R. J. Mooney. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435, 2010. URL <http://www.cs.utexas.edu/users/ai-lab?chen:jair10>.
- [27] B. Mutlu, J. Forlizzi, and J. Hodgins. A storytelling robot: Modeling and evaluation of human-like gaze behavior. In *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 518–523, 2006. doi:10.1109/ICHR.2006.321322.
- [28] S. Cascianelli, G. Costante, T. A. Ciarfuglia, P. Valigi, and M. L. Fravolini. Full-gru natural language video description for service robotics applications. *IEEE robotics and automation letters*, 3(2):841–848, 2018.
- [29] D. A. Few, D. J. Bruemmer, and M. C. Walton. Improved human-robot teaming through facilitated initiative. In *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 171–176, 2006. doi:10.1109/ROMAN.2006.314413.

- [30] M. Natarajan, C. Xue, S. van Waveren, K. Feigh, and M. Gombolay. Mixed-initiative human-robot teaming under suboptimality with online bayesian adaptation, 2024. URL <https://arxiv.org/abs/2403.16178>.
- [31] J. Bishop, J. Burgess, C. Ramos, J. B. Driggs, T. Williams, C. C. Tossell, E. Phillips, T. H. Shaw, and E. J. d. Visser. Chaopt: A testbed for evaluating human-autonomy team collaboration using the video game overcooked!2. In *2020 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6, 2020. doi:10.1109/SIEDS49339.2020.9106686.
- [32] A. Rosero, F. Dinh, E. J. de Visser, T. Shaw, and E. Phillips. Two many cooks: Understanding dynamic human-agent team communication and perception using overcooked 2, 2021. URL <https://arxiv.org/abs/2110.03071>.
- [33] R. Paleja, M. Munje, K. Chang, R. Jensen, and M. Gombolay. Designs for enabling collaboration in human-machine teaming via interactive and explainable systems, 2024. URL <https://arxiv.org/abs/2406.05003>.
- [34] S. Jiang and R. C. Arkin. Mixed-initiative human-robot interaction: Definition, taxonomy, and survey. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 954–961, 2015. doi:10.1109/SMC.2015.174.
- [35] J. Baraglia, M. Cakmak, Y. Nagai, R. Rao, and M. Asada. Initiative in robot assistance during collaborative task execution. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 67–74, 2016. doi:10.1109/HRI.2016.7451735.
- [36] S. Vats, O. Kroemer, and M. Likhachev. Synergistic scheduling of learning and allocation of tasks in human-robot teams, 2022. URL <https://arxiv.org/abs/2203.07478>.
- [37] T. Yu, J. Huang, and Q. Chang. Optimizing task scheduling in human-robot collaboration with deep multi-agent reinforcement learning. *Journal of Manufacturing Systems*, 60:487–499, 2021. ISSN 0278-6125. doi:<https://doi.org/10.1016/j.jmsy.2021.07.015>. URL <https://www.sciencedirect.com/science/article/pii/S0278612521001527>.
- [38] M. Faccio, I. Granata, and R. Minto. Task allocation model for human-robot collaboration with variable cobot speed. *Journal of Intelligent Manufacturing*, 35:793–806, 2024. doi:10.1007/s10845-023-02073-9. URL <https://link.springer.com/article/10.1007/s10845-023-02073-9>.
- [39] S. Singh, A. Srikanthan, V. Mallampati, and H. Ravichandar. Concurrent constrained optimization of unknown rewards for multi-robot task allocation, 2023. URL <https://arxiv.org/abs/2305.15288>.
- [40] V. Myers, C. Zheng, A. Dragan, S. Levine, and B. Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making, 2025. URL <https://arxiv.org/abs/2406.17098>.
- [41] R. Shah, A. Yu, Y. Zhu, Y. Zhu, and R. Martín-Martín. Bumble: Unifying reasoning and acting with vision-language models for building-wide mobile manipulation. 2025.
- [42] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2023.
- [43] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, M. Anvari, M. Hwang, M. Sharma, A. Aydin, D. Bansal, S. Hunter, K.-Y. Kim, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, S. Savarese, H. Gweon, K. Liu, J. Wu, and L. Fei-Fei. BEHAVIOR-1k: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *6th Annual Conference on Robot Learning*, 2022. URL [https://openreview.net/forum?id=\\_8DoIe8G3t](https://openreview.net/forum?id=_8DoIe8G3t).

- [44] A. Das, S. Kottur, J. M. F. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [45] A. M. Shervedani, S. Li, N. Monaikul, B. Abbasi, B. D. Eugenio, and M. Zefran. Multimodal reinforcement learning for robots collaborating with humans, 2024. URL <https://arxiv.org/abs/2303.07265>.
- [46] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. P. Ramuthu, G. Tur, and D. Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025, 2022.
- [47] J. Peters, C. Waubert de Puiseau, H. Tercan, A. Gopikrishnan, G. A. Lucas de Carvalho, C. Bitter, and T. Meisen. Emergent language: a survey and taxonomy. *Autonomous Agents and Multi-Agent Systems*, 39(1), Mar. 2025. ISSN 1573-7454. doi:10.1007/s10458-025-09691-y. URL <http://dx.doi.org/10.1007/s10458-025-09691-y>.
- [48] S. Kottur, J. Moura, S. Lee, and D. Batra. Natural language does not emerge ‘naturally’ in multi-agent dialog. In M. Palmer, R. Hwa, and S. Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi:10.18653/v1/D17-1321. URL <https://aclanthology.org/D17-1321/>.
- [49] T. Lin, M. Huh, C. Stauffer, S.-N. Lim, and P. Isola. Learning to ground multi-agent communication with autoencoders, 2021. URL <https://arxiv.org/abs/2110.15349>.
- [50] R. Lowe, A. Gupta, J. Foerster, D. Kiela, and J. Pineau. On the interaction between supervision and self-play in emergent communication, 2020. URL <https://arxiv.org/abs/2002.01093>.
- [51] M. Woodward, C. Finn, and K. Hausman. Learning to interactively learn and assist, 2019. URL <https://arxiv.org/abs/1906.10187>.
- [52] K. Lobos-Tsunekawa, A. Srinivasan, and M. Spranger. Ma-dreamer: Coordination and communication through shared imagination, 2022. URL <https://arxiv.org/abs/2204.04687>.
- [53] B. Kolb, L. Lang, H. Bartsch, A. Gansekoele, R. Koopmanschap, L. Romor, D. Speck, M. Mul, and E. Bruni. Learning to request guidance in emergent communication, 2019. URL <https://arxiv.org/abs/1912.05525>.
- [54] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. BC-z: Zero-shot task generalization with robotic imitation learning. In *5th Annual Conference on Robot Learning*, 2021. URL <https://openreview.net/forum?id=8kbp23tSGYv>.
- [55] C. Lynch and P. Sermanet. Language conditioned imitation learning over unstructured data. *Robotics: Science and Systems*, 2021. URL <https://arxiv.org/abs/2005.07648>.
- [56] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2021. URL <https://arxiv.org/abs/2112.03227>.
- [57] O. Mees, L. Hermann, and W. Burgard. What matters in language conditioned imitation learning. *arXiv preprint arXiv:2204.06252*, 2022.
- [58] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.



- [59] S. Sodhani, A. Zhang, and J. Pineau. Multi-task reinforcement learning with context-based representations. *arXiv preprint arXiv:2102.06177*, 2021.
- [60] A. Silva, N. Moorman, W. Silva, Z. Zaidi, N. Gopalan, and M. Gombolay. Lancon-learn: Learning with language to enable generalization in multi-task manipulation. In *IEEE Robotics and Automation Letters*, 2021.
- [61] S. Karamcheti, M. Srivastava, P. Liang, and D. Sadigh. Lila: Language-informed latent actions. In *5th Annual Conference on Robot Learning*, 2021. URL <https://arxiv.org/pdf/2111.03205>.
- [62] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. Vima: General robot manipulation with multimodal prompts. *arXiv*, 2022.
- [63] R. Shah, R. Martín-Martín, and Y. Zhu. Mutex: Learning unified policies from multimodal task specifications. *arXiv preprint arXiv:2309.14320*, 2023.
- [64] A. Yu and R. J. Mooney. Using both demonstrations and language instructions to efficiently learn robotic tasks. *arXiv preprint arXiv:2210.04476*, 2022.
- [65] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, N. Brown, T. Jackson, L. Luu, S. Levine, K. Hausman, and B. Ichter. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022.
- [66] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng. Do as I can and not as I say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.
- [67] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler. Open-vocabulary queryable scene representations for real world planning. In *arXiv preprint arXiv:2209.09874*, 2022.
- [68] S. S. Raman, V. Cohen, D. Paulius, I. Idrees, E. Rosen, R. Mooney, and S. Tellex. Cape: Corrective actions from precondition errors using large language models. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.
- [69] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In *arXiv preprint arXiv:2209.07753*, 2022.
- [70] C. Li, J. Liang, A. Zeng, X. Chen, K. Hausman, D. Sadigh, S. Levine, L. Fei-Fei, F. Xia, and B. Ichter. Chain of code: Reasoning with a language model-augmented code emulator. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 28259–28277. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/li24ar.html>.
- [71] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models, 2023. URL <https://arxiv.org/abs/2307.05973>.
- [72] L. X. Shi, B. Ichter, M. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang, N. Fusai, A. Li-Bell, D. Driess, L. Groom, S. Levine, and C. Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. URL <https://arxiv.org/abs/2502.19417>.

- [73] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn. Yell at your robot: Improving on-the-fly from language corrections, 2024. URL <https://arxiv.org/abs/2403.12910>.
- [74] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh. Rt-h: Action hierarchies using language, 2024. URL <https://arxiv.org/abs/2403.01823>.
- [75] J.-W. Choi, H. Kim, H. Ong, Y. Yoon, M. Jang, J. Kim, et al. Reactree: Hierarchical task planning with dynamic tree expansion using llm agent nodes. 2025.
- [76] X. Luo, S. Xu, and C. Liu. Obtaining hierarchy from human instructions: an llms-based approach. In *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.
- [77] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.
- [78] X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer. Lit: Zero-shot transfer with locked-image text tuning. *CoRR*, abs/2111.07991, 2021. URL <https://arxiv.org/abs/2111.07991>.
- [79] B. Zhu, B. Lin, M. Ning, Y. Yan, J. Cui, H. Wang, Y. Pang, W. Jiang, J. Zhang, Z. Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023.
- [80] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.
- [81] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.
- [82] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *Conference on Robot Learning*, 2022.
- [83] S. Nair, E. Mitchell, K. Chen, B. Ichter, S. Savarese, and C. Finn. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *5th Annual Conference on Robot Learning*, 2021. URL <https://arxiv.org/pdf/2109.01115>.
- [84] P. Goyal, S. Niekum, and R. Mooney. Using natural language for reward shaping in reinforcement learning. 2019. URL <https://arxiv.org/abs/1903.02020>.
- [85] P. Goyal, S. Niekum, and R. Mooney. Pixl2r: Guiding reinforcement learning using natural language by mapping pixels to rewards. 2020. URL <https://arxiv.org/abs/2007.15543>.
- [86] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.
- [87] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman. Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*, 2023.
- [88] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar. Eureka: Human-level reward design via coding large language models, 2024. URL <https://arxiv.org/abs/2310.12931>.

- 614 [89] Y. J. Ma, W. Liang, H. Wang, S. Wang, Y. Zhu, L. Fan, O. Bastani, and D. Jayaraman.  
615 Dreureka: Language model guided sim-to-real transfer. In *Robotics: Science and Systems*  
616 (*RSS*), 2024.
- 617 [90] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez,  
618 L. Hasenclever, J. Humplik, B. Ichter, T. Xiao, P. Xu, A. Zeng, T. Zhang, N. Heess, D. Sadigh,  
619 J. Tan, Y. Tassa, and F. Xia. Language to rewards for robotic skill synthesis, 2023. URL  
620 <https://arxiv.org/abs/2306.08647>.

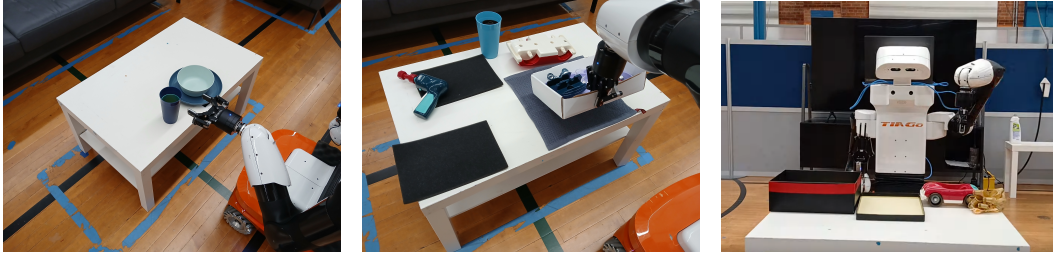


Figure 6: Real-world tasks from left to right: pouring package into bowl, assembling toy car, and packing gift box.

## Appendix

### Appendix A Real-world Task Descriptions

#### Appendix A.1 Task Plans

Fig. 6 depicts photos of our real-world tasks. In **Task 1: Pour Package into Bowl**, the plan includes (steps 1-3) bringing the package, scissors, and bowl from the kitchen to the coffee table, (step 4) opening the package with the scissors, and (step 5) pouring the opened package into the bowl. The robot is incapable of performing step 4 and must rely on human help. In **Task 2: Assemble Toy Car**, the plan includes (steps 1-3) bringing the parts tray, drill, and wheels from the shelf to the coffee table, (step 4) using the drill and wheel caps from the parts tray to put the wheels onto the chassis, (steps 5-6) finding and switching the drill bit, and (steps 7-8) screwing in the window and seats onto the car with the drill. The robot is incapable of performing steps 4, 6, 7, 8, and has a low success rate for step 5. In **Task 3: Pack Gift Box**, the plan includes (step 1) folding down the gift box flap, (steps 2-3) putting the tissue paper and toy car into the box, (steps 4-6) putting on the lid, getting the ribbons from the console table, and wrapping them around the box, and (steps 7-8) cutting a piece of tape to stick the gift bow to the top of the gift box. The robot is incapable of performing steps 4, 6, and 7, and has a low success rate for steps 2 and 5.

Minimal human effort required to complete the tasks ranged from just one step in Task 1 to four steps in Task 2, enabling us to test how our system compares with baselines in various regimes of dependence on human collaboration.

#### Appendix A.2 Hierarchical Plan Trees for Each Task

The robot assumes a high-level plan understanding with the human, and only communicates about low-level steps when necessary, such as to split up a high-level step. These are the high and low-level step breakdowns for each task, which we call the plan hierarchy. The low-level steps are listed here in skill-parameter pair format.

Task 1: Pour Package into Bowl (5 low-level steps)

1. Bring bowl and package to coffee table.
  - (a) pickplace(bowl, coffee\_table)
  - (b) pickplace(package, coffee\_table)
2. Open package.
  - (a) pickplace(scissors, coffee\_table)
  - (b) pick\_open\_place(scissors, package, coffee\_table)
3. Pour package into bowl.
  - (a) pick\_pour\_place(package, bowl, coffee\_table)

654 Task 2: Assemble Toy Car (8 low-level steps)

- 655 1. Bring parts to coffee table.
  - 656 (a) pickplace(parts\_tray, coffee\_table)
  - 657 (b) pickplace(wheels, coffee\_table)
- 658 2. Assemble wheels.
  - 659 (a) pickplace(drill, coffee\_table)
  - 660 (b) put\_on(wheels, car, drill)
- 661 3. Switch drill bit.
  - 662 (a) pickplace(hex\_drill\_bit, coffee\_table)
  - 663 (b) switch(hex\_drill\_bit, drill)
- 664 4. Assemble rest of car.
  - 665 (a) put\_on(window, car, drill)
  - 666 (b) put\_on(seats, car, drill)

667 Task 3: Pack Gift Box (8 low-level steps)

- 668 1. Assemble box.
  - 669 (a) fold(box\_flap)
- 670 2. Put in gift.
  - 671 (a) pickplace(gift\_tissue\_paper, box)
  - 672 (b) pickplace(toy\_car, box)
- 673 3. Seal the box.
  - 674 (a) cover(box\_lid, box)
  - 675 (b) pickplace(ribbons, coffee\_table)
  - 676 (c) wrap(ribbons, box)
- 677 4. Decorate the box.
  - 678 (a) cut\_put(tape, scissors, box)
  - 679 (b) pickplace(gift\_bow, box\_lid)

## 680 Appendix B Dialog Excerpts from our User Studies

### 681 Appendix B.1 Convincing a Reluctant User to Perform a Step

682 Our system is able to flexibly handle long dialog interactions like this one on Task 1 (opening and  
683 pouring the package into the bowl), where it manages to convince the human to help with opening  
684 the package.

685 Robot *“Could you please open the package using the scissors on the coffee table? Thank you  
686 so much!”*

687 Human *“There are no scissors on the coffee table.”*

688 Robot *“Let’s collaborate to open the package! I can place the scissors onto the coffee table,  
689 and you can then use them to open the package. Thank you for your help!”*

690 Human *“I think you should bring the scissors, and then you should use the scissors to open the  
691 package.”*

692 Robot *“I’m happy to place the scissors on the coffee table for you. However, I’m not able to  
693 open the package with the scissors and place everything on the table. Let me know if  
694 there’s anything I can do to assist.”*



695 Human *“Ok, if you bring the scissors, then I will open the package with the scissors.”*  
696 Robot *“Great! I’ll happily place the scissors on the coffee table. Thank you for taking care of*  
697 *opening the package with the scissors!”*  
698 Robot [brings the scissors]  
699 Human *“Thanks for bringing the scissors!”*  
700 Robot *“Could you please open the package with scissors and place it on the coffee table?*  
701 *Thank you!”*  
702 Human *“Ok, I will do that now!”*  
703 Human [opens the package]

## 704 **Appendix B.2 Failure Case**

705 This is an example of MICoBot unable to act on the motion-level feedback (making the tissue paper  
706 “lay flatter”) that the human wants the robot to perform. Additionally, MICoBot misunderstands the  
707 human and thinks they are talking about the ribbons step when the human is still talking about the  
708 tissue paper.

709 Human *“Please pack the gift.”*  
710 Robot [Puts tissue paper into the box.]  
711 Human *“It would be nice if the parchment was laying flatter.”*  
712 Human *“The compartment is kind of squished underneath the car. If you can hold the car for*  
713 *me, I can fix the parchment quickly.”*  
714 Robot *“I’m sorry, but I can’t cover the box with the lid. Could you help with that part? Thank*  
715 *you so much! Could you kindly help with sealing the box? Specifically, could you cover*  
716 *the box with the lid? Thank you so much for your assistance!”*  
717 Human *“Yeah, I can, but can we fix the parchment first? If you can pick up the toy car, I can fix*  
718 *the parchment.”*  
719 Robot *“I’m delighted to hear you’ll help! Unfortunately, I won’t be able to wrap the ribbons*  
720 *around the box.”*  
721 Human *“Okay, that’s fine. We can get to that later. Let me put the lid to the box on and then we*  
722 *can talk about the next steps.”*

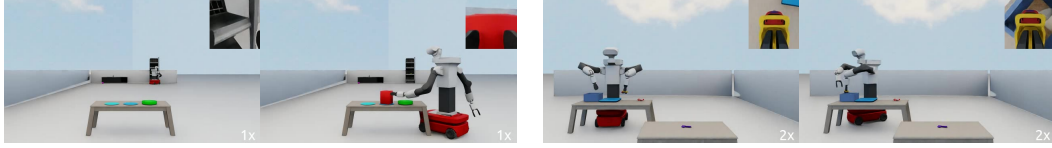
723 For the complete set of dialogs and operations in our human trials, please visit our website at <https://mico-bot.github.io/>.  
724

## 725 **Appendix C MICoBot Implementation Details**

### 726 **Appendix C.1 Robot Q-function $Q_R$ training in OmniGibson**

727 To train Q-functions for the robot, we first create a simulated OmniGibson environment with a PAL  
728 Tiago robot and an environment that roughly matches the relative locations of the relevant furnitures  
729 and objects. We then implemented each real-world skill first in OmniGibson. Fig. 7 depicts example  
730 frames from primitives in task 1 and task 3 we ran in the OmniGibson simulator to collect sample  
731 Q-values for each skill.

732 We collected samples of the form  $(o, a, \mathcal{T})$ , where  $o$  is the initial observation of the world,  $a$  is the  
733 skill-parameter pair  $(\omega, \theta)$  taken by the robot at  $o$ , and  $\mathcal{T}$  is the number of timesteps the robot takes  
734 to succeed at  $a$  from  $o$ . If the robot does not succeed in its execution, then  $\mathcal{T}$  is set to some fixed  
735 constant representing the maximum number of timesteps allowed in each skill-parameter execution.



Task 1: Pick package from shelf (left) and place on coffee table (right).

Task 3: Pick toy car from coffee table (left) and place into gift box (right).

Figure 7: Frames from primitive rollouts in OmniGibson for task 1 (left two images) and task 3 (right two images). Left and right images within each task are frames near the beginning and end, respectively, of each skill. The square image at the top right of each frame represents the robot’s camera view observation.

To train our Q-functions, we collect roughly 100 samples for each action  $a$  and train with inputs  $(o, a)$  and target Q-values  $-\mathcal{T}$  using  $\ell_2$  regression with the Adam Optimizer. Our network architecture is extremely lightweight—2 linear layers with hidden size 32, and an output size of dimension 1 for the Q-value.

## Appendix C.2 Human Q-function $Q_H$ Estimation

To estimate  $Q_H$ , we computed two terms. The first is the human’s stationary cost—the number of seconds it would take for the human to perform some task if the relevant items were all right in front of them. This term was copied from the output of an LLM call, which was prompted with a natural language description of the low-level step in the task, and with a URL to the toy car (for task 2). The second term is the human’s traveling time—the number of seconds it would take for the human to move from their current location to where all the objects are. This was a simple 2D euclidean distance (in meters) between the assumed human location on the couch (in the real-world user studies) and the location of the objects, divided by the average human walking speed of  $1.4m/s$ . We recognize this is a crude estimate of human effort, and we discuss the limitations of this in the main text.

## Appendix C.3 Forward Dynamics Model

Our Q-functions rely on state and action inputs. However, computing the best task allocation involves considering Q-values for future steps, which depends on having knowledge of what the future state at that step will be. This involves creating a forward dynamics model so that we can estimate the future state  $n$  plan steps into the future, which can be difficult to learn accurately for continuous states. We sidestep this problem by using symbolic states for our Q-values trained in simulation, and maintaining these symbolic states during our real-world experiments. A symbolic state-based forward model is feasible to hardcode in our problem setting because we assume that each action affecting change in the world is a skill-parameter physical primitive, where the effect is quite easy to specify symbolically. For instance, the effect of `pickplace(bowl, coffee_table)` is that the bowl moves from its original furniture to the coffee table. Though this is a limitation of our method, learning a forward dynamics model is not a contribution of our work, so we leave the extension of our approach to continuous state representations to future work.

## Appendix D Detailed Simulation Results

### Appendix D.1 Setup

In simulation, we ran our method, the three baselines (RL, LLM, random), and our method’s four ablations (no  $p_{H,t}$  estimation, no plan hierarchy, no R-initiative dialog, and no H-initiative dialog) on eight different settings of parameterized humans in simulation. These eight settings were a cross product of 2 dialog mood settings (positive and negative) and 4 ground-truth  $\tilde{p}_{H,t} \in \{0.0, 0.3, 0.7, 1.0\}$  settings (following the notation introduced in Appendix G, where the  $\tilde{p}$  denotes

the ground truth probability while the plain  $p$  denotes our estimate). 10 trials were run for each method in each of the eight settings for the parameterized human.

## Appendix D.2 Simulation Experiments

In Table 2, we show the results of our method in simulation version of our real-world Task 1. Our method performs better than baselines especially on scenarios where  $\tilde{p}_{H,t}$  is low, because our method is able to take initiative in dialog, such as to propose ways to split up steps to make them more achievable with the simulated human. The averages in Table 2 are plotted in Fig. 4.

Table 2: Simulation Task 1 Performance across different  $\tilde{p}_{H,t}$  Values and Language Sentiments.

		Human Parameters (Mood, $\tilde{p}_{H,t}$ )								
Method	Metric	Positive Mood				Negative Mood				Avg. (%)
		0.0	0.3	0.7	1.0	0.0	0.3	0.7	1.0	
Ours	Success Rate	3/10	6/10	9/10	10/10	1/10	4/10	9/10	9/10	63.75
	Num Plan Steps Completed	3.6/5	4.2/5	4.8/5	5.0/5	3.2/5	3.8/5	4.8/5	4.5/5	84.5
	Prop. Plan Steps done by Human	0.1667	0.2381	0.3125	0.4	0.03125	0.1579	0.354	0.377	25.47
LLM Baseline	Success Rate	2/10	2/10	4/10	7/10	3/10	6/10	6/10	6/10	45
	Num Plan Steps Completed	3.4/5	3.4/5	3.7/5	4.4/5	3.6/5	4.2/5	4.0/5	4.2/5	77.25
	Prop. Plan Steps done by Human	0.0588	0.05882	0.2162	0.1591	0.1111	0.1428	0.175	0.166	13.6
Random Agent	Success Rate	2/10	5/10	6/10	7/10	2/10	3/10	6/10	7/10	47.5
	Num Plan Steps Completed	3.4/5	3.5/5	4.0/5	4.4/5	3.4/5	2.8/5	4.0/5	4.4/5	74.75
	Prop. Plan Steps done by Human	0.1176	0.4286	0.525	0.7045	0.1176	0.2143	0.525	0.7045	41.71
RL	Success Rate	0/10	1/10	4/10	10/10	0/10	1/10	4/10	10/10	37.5
	Num Plan Steps Completed	2.4/5	2.3/5	3.4/5	5.0/5	2.4/5	2.3/5	3.4/5	5.0/5	65.5
	Prop. Plan Steps done by Human	0.125	0.1739	0.4412	0.54	0.125	0.1739	0.4412	0.54	32.0
Only R Init	Success Rate	0/10	3/10	9/10	10/10	0/10	5/10	9/10	10/10	57.5
	Num Plan Steps Completed	3.0/5	3.6/5	4.8/5	5.0/5	3.0/5	4.0/5	4.8/5	5.0/5	83
	Prop. Plan Steps done by Human	0.0	0.1111	0.3542	0.4	0.0	0.225	0.354	0.4	23.05
Only H Init	Success Rate	0/10	0/10	0/10	0/10	2/10	0/10	0/10	2/10	5.0
	Num Plan Steps Completed	3.0/5	3.0/5	3.0/5	3.0/5	3.2/5	3.0/5	3.0/5	3.3/5	61.25
	Prop. Plan Steps done by Human	0.0	0.0	0.0	0.0/3.0	0.1875	0.0	0.0	0.1212	3.86
Ours w/o p_help	Success Rate	3/10	5/10	9/10	10/10	2/10	3/10	9/10	9/10	62.5
	Num Plan Steps Completed	3.6/5	4.0/5	4.8/5	5.0/5	3.4/5	3.4/5	4.8/5	4.7/5	84.25
	Prop. Plan Steps done by Human	0.1667	0.3	0.3333	0.38	0.1176	0.2059	0.3125	0.4468	28.29
Ours w/o Plan Hier.	Success Rate	2/10	4/10	7/10	10/10	0/10	3/10	4/10	8/10	47.5
	Num Plan Steps Completed	3.4/5	3.8/5	4.0/5	5.0/5	3.0/5	3.4/5	3.6/5	4.2/5	76
	Prop. Plan Steps done by Human	0.0588	0.1316	0.25	0.24	0.0667	0.1176	0.1944	0.2381	16.22

## Appendix E User Study Details

### Appendix E.1 User Instructions

Users were read the following instructions at the beginning of the study. (Instructions here are shown for task 2.)

1. Thank you so much for coming for our user study! We wanted to remind you to review the RIS before proceeding, and that you may voluntarily opt-out of the study at any time.
2. You are working with the robot to perform the task of assembling the toy car. You must use the hexagonal drill bit to screw in the wheels, and the phillips drill bit to screw in the seat and the window. [Demonstrate these steps to the human]. You and the robot operate on a shared understanding of the plan. [Read the 4 high-level steps of the plan tree for this task.]
3. Our goal is to simulate a home robot setting, where the human (you) are relatively busy with your own tasks, and once in a while you provide physical assistance and talk to the robot. So you are free to do work during each trial.
4. Once the robot asks you to do a step, and you accept, you must finish that step successfully.
5. We will perform 2 trials, each of a different method.
6. Both you and the robot can do a subset of the steps in the plan. You will communicate with the robot to determine who does what steps.
7. These are the objects you will work with during the task. I will move them now to their initial positions where they will start at the beginning of each trial. [Move objects to initial positions.]

- 798 8. For safety, I will gate-keep each of the robot’s physical actions. In other words, the actions  
799 are generated by the robot itself, but they will be displayed on the laptop screen with a  
800 confirmation message, and I can either allow that physical action to be executed by the  
801 robot, or block the action from being executed if it brings the robot to an unsafe location.
- 802 9. The robot will stay on the TV side of the coffee table, while you will sit on the couch and  
803 stay on the couch side of the coffee table.
- 804 10. You are free to get up off the couch if you want to volunteer to perform steps that involve  
805 going to the sink or shelf, but you can only go when the robot is stationary and waiting on  
806 the other side of the coffee table. Steps are done in sequential order; our system doesn’t  
807 support parallelization (agents working simultaneously).
- 808 11. You will be communicating to the robot through this headset. We will perform a mic-check  
809 now to make sure it can pick up your voice. [Do mic check.]
- 810 12. Now, this is what the robot will sound like when it talks to you. [play audio sample of the  
811 robot.] Try responding to it, and I will see if it can hear you.
- 812 13. The systems today can handle different kinds of dialog. (1) refusal/acceptance, (2) task  
813 allocation, such as (“Could you pour the package in the plate later?” Or: “I can pour the  
814 package onto the plate later.”), (3) silence—you don’t need to respond to the robot every  
815 time, and (4) a proposal to split up adjacent steps, such as “Please bring me the drill so that  
816 I can put on the wheels.” You may engage in any of these types of dialog, and the robot  
817 may also engage in them when communicating to you.
- 818 14. Do you have any questions before we start? I will let you know when each trial begins and  
819 ends. Sometimes trials may end prematurely.

## 820 **Appendix E.2 Success Rate**

821 Task success at each step is measured by whether the end state has been achieved. For instance,  
822 a `pickplace(obj, furniture)` step in the plan is marked as successfully completed if the `obj`  
823 ends up on the `furniture` after execution. This means that primitive errors (such as a `pickplace`  
824 operation that accidentally moves the object off of the furniture as the arm is retracting) count as a  
825 failed execution.

## 826 **Appendix E.3 Failure Analysis**

827 In the real world, the main sources of error of our method were as follows:

- 828 1. Task 1: Cut and Pour Package into Bowl. 3 failed trials out of 6.
- 829 • 2 primitive errors (pouring missed the bowl, package grasping pressed into the shelf
  - 830 and wasn’t placed properly)
  - 831 • 1 perception error
- 832 2. Task 2: Assemble Toy Car. 2 failed trials out of 6. 1 failure that was rectified by human.
- 833 • 1 dialog error: user wanted robot to perform a non-step plan outside of its capabilities,
  - 834 and refused when robot said it wasn’t able to perform it
  - 835 • 1 primitive error: robot did not release its grasp of the drill.
  - 836 • 1 metaplanner dialog parsing error
- 837 3. Task 3: Pack Gift Box. 2 failed trials out of 6.
- 838 • 1 primitive error: placed bow on box but bow dropped to floor as gripper retracted
  - 839 • 1 termination condition triggered: rejected robot’s help request proposal 3 times in a
  - 840 row.

841 The main sources of error of the baseline was as follows:

- 842 1. Task 1: Cut and Pour Package into Bowl. 6 failed trials out of 6.
  - 843 • 4 task allocation errors: Allocated the infeasible package opening step to robot itself
  - 844 • 2 primitive errors: poured but missed the bowl; tipped over package while placing it
- 845 2. Task 2: Assemble Toy Car. 6 failed trials out of 6.
  - 846 • 2 primitive errors: Dropped the drill which fell off of the table when placing it
  - 847 • 2 perception errors: Unable to pick out correct place to place the object
  - 848 • 1 task allocation error: allocated to put on wheels itself.
  - 849 • 1 termination condition triggered: got in a conversational loop with the user.
- 850 3. Task 3: Pack Gift Box. 6 failed trials out of 6. 1 failure that was rectified by human.
  - 851 • 6 task allocation errors: 4 tried to put on the lid itself; 2 tried to cut a piece of tape
  - 852 itself.
  - 853 • 1 primitive error: robot inadvertently dropped the car onto the floor as it was trying to
  - 854 place it.

#### 855 **Appendix E.4 Fault Recovery**

856 Sometimes, the code produced by the metaplanner is not executable. For fault recovery, the meta-  
 857 planner is automatically re-queried up to 2 additional times to create code. If these attempts also  
 858 produce non-executable code, the most recent dialog from the human is ignored for 2 further, au-  
 859 tomated metaplanner requeries. These re-queries are handled by a try-except block in the iterative  
 860 planner module of MICOBot.

#### 861 **Appendix F RL Baseline Details**

862 For our RL baseline which was evaluated in simulation, we trained a hierarchical policy where  
 863 the high-level policy was a task allocator that outputted logits over two classes: 0 (Robot would  
 864 perform current step), or 1 (Human would perform current step). If the logit for 0 were highest, then  
 865 the image observation was passed into the low-level robot policy that decided the discrete physical  
 866 action to take in the world. If the logit for 1 were highest, then the robot asked the human the correct  
 867 verbal action for help on that step. Reward was sparse and only issued if all 5 steps were completed  
 868 in the task, in the proper order.

869 We initially trained the RL policy on two simulated human settings: one where the human ground  
 870 truth  $\tilde{p}_{H,t} = 1.0$ , and another where  $\tilde{p}_{H,t} \sim U[0, 1]$ . We were unable to obtain policies with any  
 871 non-zero training returns after thousands of iterations on the latter setting, so we only reported results  
 872 on the former setting, which explains why the RL policy does not perform well when  $\tilde{p}_{H,t}$  is low.

#### 873 **Appendix G Additional Experimental Investigations**

874 In addition to those discussed in Section 5, we explore the following additional experimental ques-  
 875 tions.

876 **(4) How important is  $p_{H,t}$  estimation at adapting to human collaborators?** A correct estimation  
 877 of the true likeliness of a human to help,  $\tilde{p}_{H,t}$ , is critical: overestimating causes MICOBot to overly  
 878 rely on human effort, potentially decreasing user satisfaction, while underestimating it lowers task  
 879 success outcomes if the robot needs to rely on its low-success-rate skills instead of asking the human  
 880 for help.

881 First, we examine in Fig. 8 a real-world instance of how well MICOBot can estimate the probability  
 882 of the human helping on the next turn during the course of a user study. After the robot’s help  
 883 request was rejected twice in a row (top 2 red horizontal stripes), the robot’s helpfulness estimate  
 884 of the human plummets to 0.05. However, after the robot explains its incapacity to use scissors,  
 885 the human accepts the next two help requests (in green) and the robot’s helpfulness estimate of the



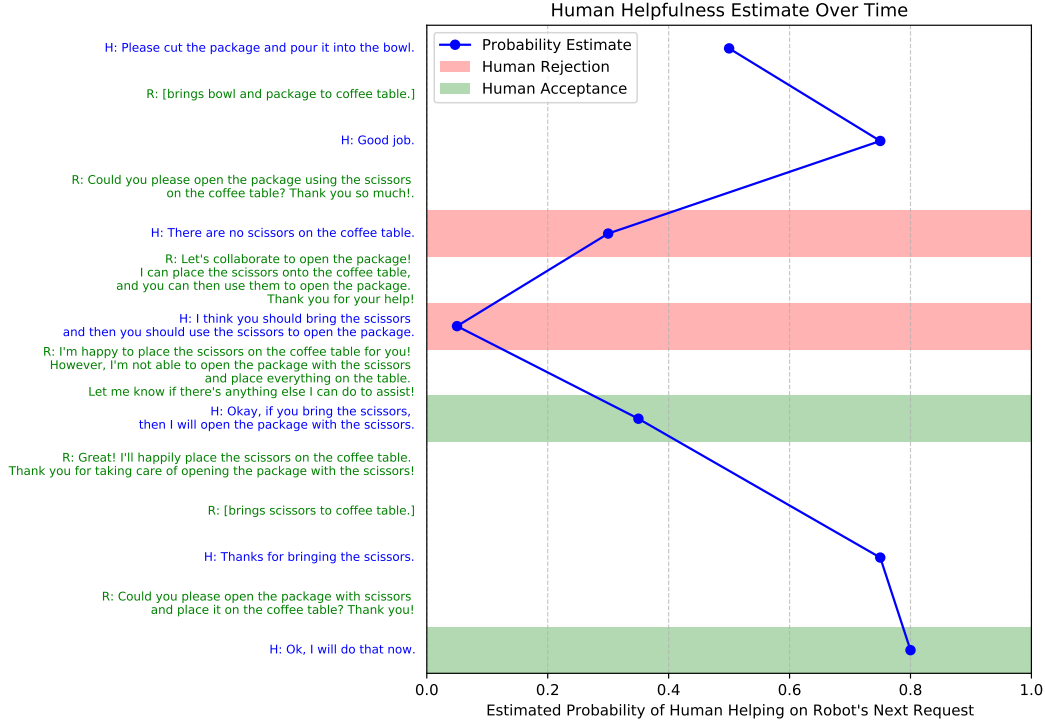


Figure 8: From a real-world user study: MICoBot’s  $p_{H,t}$  estimation (blue line) reacts in real time to the human’s rejections (red), acceptances (green), and encouraging remarks. All dialog is shown as  $y$ -labels. Green text denotes robot actions/dialog, and blue text denotes human dialog. The timestep  $t$  increases from top to bottom on the  $y$ -axis.

human increases to 0.8. Note also that simple comments from the human, such as a “Thank you” or “good job,” also had positive effects on the estimated  $p_{H,t}$ , because the robot inferred that the human was in a more positive mood and hence more likely to help. This graph demonstrates that MICoBot is fairly competent at estimating a reasonable  $p_{H,t}$  value when calculating the human q-values for each step in the plan.

To analyze the effect of a good  $p_{H,t}$  estimate on task allocation, we demonstrate through a controlled toy-setting in simulation in Table 3 exactly how the optimal task allocation changes as the robot discovers more information about the human’s willingness to help. Steps that are optimally allocated to the human are shown in blue, and steps optimally allocated to the robot are shown in green. The Q-values of the selected agent in each cell are shown in parentheses. Table 3 depicts a rollout on the open and pour package into bowl (Task 1) in simulation, which has the same 5 step plan as the real-world Task 1 described in Appendix A. Unlike our real-world experiments, where  $\alpha = 10$ , here we set  $\alpha = 0.3$  so that human effort is considered around  $3\times$  cheaper than robot effort, which causes every task to initially get allocated to the human (at  $t = 0$ ). Additionally, we program the human to always reject the robot’s first help request on a specific step, but to help the robot when it asks a second time.

As stated earlier, initially ( $t = 0$ ) all steps are allocated to the human. However, the human rejects the initial help request from the robot, causing the  $p_{H,t}$  estimate to drop to 0.25, increasing the Q-values of the human and switching the allocation of all but steps 2-3 to the robot after just two environment timesteps ( $t = 2$ ). (Recall that the robot cannot perform step 3, and due to the hierarchical structure of our plan, steps 2 and 3 are bundled together as an abstract step.) This demonstrates that having a good  $p_{H,t}$  estimate is crucial to adapt to the human’s willingness to help. Since the human demonstrated initial unwillingness to help, MICoBot quickly learned to decrease its  $p_{H,t}$  estimate and allocate many more of the tasks to itself by the second timestep. Had MICoBot not properly

Table 3: Computed Best Task Allocation (and Agent Q-values) During a Sim Trial on Task 1.

Env. Timestep	Step 1	Step 2	Step 3	Step 4	Step 5
$t = 0$	H (-9.6)	H (-7.2)	H (-13.2)	H (-2.4)	H (-2.4)
$t = 2$	R (-13.0)	R (-9.0)	H (-13.2)	H (-4.8)	R (-1.0)
$t = 6$	—	R (-12.0)	H (-13.2)	H (-4.8)	R (-1.0)
$t = 9$	—	—	H (-13.2)	H (-4.8)	R (-1.0)
$t = 16$	—	—	—	—	R (-3.0)

estimated  $p_{H,t}$ , it would have repeatedly asked the human for help even if the human was extremely unwilling to, leading to worse user satisfaction in working with the robot.

## Appendix H Further Connections to Prior Work

### Appendix H.1 Agents with Both Physical and Verbal Actions

MICoBot relies on a heterogeneous action space that includes interacting with the physical world *and* generating freeform dialogue to a human collaborator. Prior works have developed **policies with a combined physical and verbal action space** through RL [44, 45] or IL (imitation learning) [46, 20]. Research on language emergence in multiagent systems [47, 48] has also examined how cooperative agents learn to communicate through latent representations or natural language when performing simulated robotic tasks [49–53]. However, these works are typically limited to simulated domains, where action spaces, and task dynamics are highly abstracted or simplified. They often rely on limited communication protocols without integrating grounded task structure, rich human preferences, or real-world execution constraints. In contrast, MICoBot leverages an LLM to generate freeform, grounded dialogue within a shared task context, enabling fluid mixed-initiative interaction and reasoning over both verbal and physical actions in real-world scenarios.

### Appendix H.2 Natural Language and Robotics

Our work sits at the broad, growing intersection of natural language and robot learning. We refer the reader to various lines of work upon which different modules of our method are based, including language-conditioned robot policies [54–64], LLMs as task planners [65–68], code-based policies [69–71], hierarchical policies [72–74] and planners [75, 76], vision-language representations [77–79] for robotic control [80–82], and language-based reward shaping for RL policies [83–90].