Mixed Initiative Dialog for Human-Robot Collaborative Mobile Manipulation

Albert Yu, Chengshu Li, Luca Macesanu, Arnav Balaji, Ruchira Ray, Ray Mooney, Roberto Martín-Martín
UT Austin, Stanford

***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/**

I. Introduction

We aim to build robots that can seamlessly collaborate with humans in everyday household tasks. Such collaboration is critical for deploying today's robots with limited capabilities. To be a truly effective partner, a collaborative robot must strive for task success with minimal human effort, while adapting dynamically to a human user's capabilities, preferences, and willingness to help. Beyond adapting, the robot must also learn to communicate the real-time delegation of task components based on which agent—human or robot—is better suited for each. Existing approaches fall short of this ideal: modern AI assistants respond only to human-initiated interactions [58, 1], while prior human-robot interaction (HRI) solutions often assume full control over the collaboration plan and complete willingness from the human partner [67]. We argue that a truly collaborative human-robot team requires a paradigm shift towards a model where both agents can take initiative to propose, bargain, and accept or reject proposals from each other as they discuss in natural language how to best complete a task. In this paper, we introduce such a collaborative system, MICoBot (Mixed-Initiative Collaborative roBot), which we believe is the first to enable mixed-initiative natural language dialog for real-world physical collaboration between robots and humans.

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

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

In summary, our work's contributions are four-fold: (1) we introduce a **new problem setting** that integrates mixed-initiative natural language dialog with mixed-initiative human-robot interaction; (2) we propose a novel **optimization framework** for task allocation, balancing human and robot effort and success through a unified metric; (3) we provide a **new simulator** for collaborative household tasks built on top of MiniBehavior [32] that includes LLM-controlled virtual humans and is available on our website; and (4) **robotic system and framework**: we develop MICoBot, a three-

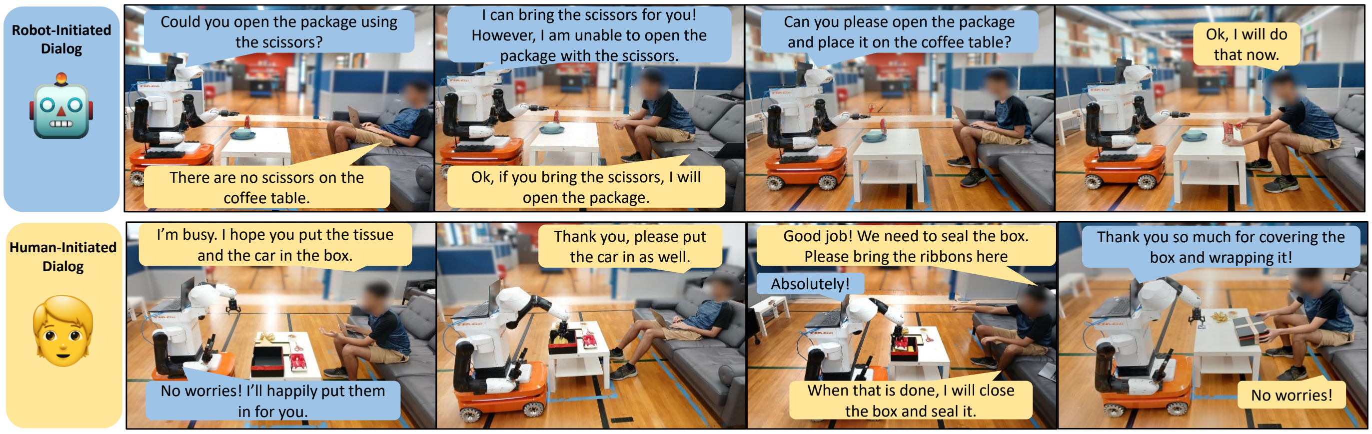


Fig. 1: We present MICoBot, a system for human-robot collaborative settings where both agents can initiate and carry out physical and verbal actions to negotiate how to accomplish a task together. Our system supports *both* robot-initiated (top row) *and* human-initiated (bottom row) task-directed dialog, 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).

level hierarchical solution for mixed-initiative speech2speech human-robot collaboration that flexibly adapts to a wide range of real human collaborators in physically grounded, long-horizon tasks.

II. Related Work

Mixed-initiative dialog [8, 3, 15] refers to communication with freeflowing questions and answers from both parties. In the NLP field, the dominant chatbot paradigm adopted by large language models (LLMs) largely eschews mixed-initiative interaction: humans pose substantive questions, and the chatbot primarily responds to fulfill these requests [58, 1]. Recent work has sought to make dialog systems more goal-directed and proactive by incorporating mixed-initiative strategies—for example, persuading users to donate to charity, enhancing users' emotional well-being [17, 88, 12, 19], or clarifying ambiguous human requests [62, 18, 13]. However, none of these systems addressed mixed-initiative dialog in grounded, real-world collaborative scenarios involving physical manipulation tasks.

In the human-robot interaction (HRI) field, researchers have developed **human-robot collaboration systems** that interact through language but are restricted to **single-initiative dialog**. Some of these systems integrate LLMs as task planners or delegators [83, 50, 22] for tasks like real-world cooking [83] and object sorting [50]. Other systems implement a leader-follower paradigm in simulated worlds, where the leader issues natural language instructions for the follower to execute [79, 35, 80, 24]. Single-initiative HRI systems can ask humans for clarification [65] or assistance [6, 34, 82], or inform humans of their observations [11, 53, 9]. However, by supporting only single-initiative dialog, these systems lack the capacity to adapt to the evolving nature of the human, robot, and environment—limiting their capacity to find the optimal division of labor that respects user preferences. [50].

Some works in HRI have explored **mixed-initiative collaborative systems without dialog**, only with physical actions [23, 57, 7, 66, 60, 30]. In particular, Baraglia et al. [4] studied separate regimes of agent initiative (human-initiative, requesting help, or robot-initiative, proactively helping), but failed to support a natural human-robot dialog. By focusing solely on physical actions, these prior works overlook the critical role of communication in effective collaboration, thereby limiting the flexibility of the human-robot team. With MICoBot, we enable both agents to take initiative—through both physical and verbal actions—via task-grounded dialog.

Several prior works in robotics and planning have studied the problem of **human-robot optimal task allocation**, typically optimizing the time to perform a task or minimizing idle agents, posing the problem as a scheduling problem [81, 86]. Others have prioritized different objectives, such as safety [20], through the formulation of a constrained optimization problem [77]. While these solutions may achieve shorter execution times, they assume a priori known capabilities and availability of all agents, including both robots and humans. In contrast, MICoBot can adapt to the specific human's willingness to help by estimating its availability based on previous dialog.

III. Problem Setting: Task Collaboration with Mixed-Initiative Dialog

MDP Formulation. In this paper, we study human-robot collaboration for shared manipulation problems where mixed-initiative

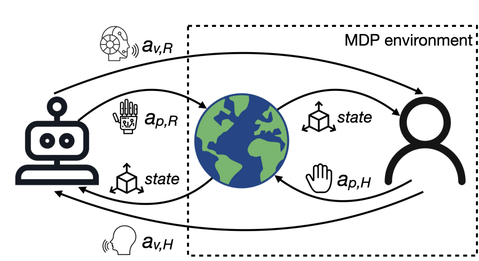


Fig. 2: Our MDP Formulation for Mixed-Initiative Collaboration

dialog occurs.
In these problems, we assume that both agents can observe the state of the world, $s \in \mathcal{S}$, and perform actions, $a \in \mathcal{A} = \mathcal{A}_p \cup \mathcal{A}_v$, comprised of a physical action space, $\mathcal{A}_p$, that directly affects the physical state of the environment s, (e.g., move objects, open them, etc.), and a free-form, natural language verbal action space, $\mathcal{A}_v$, that are directly observed by the other agent but do not change the physical state. We model the problem as a Markov Decision Process (MDP) from the robot's point of view (see 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 receives an observation $o = [I, a_{v,H}, s_{proprio}]$ consisting of an RGB-D image I, an optional verbal action from the human partner $a_{v,H}$, and the robot's proprioceptive state $s_{proprio}$. Within each environment 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 and verbal action space after perceiving the world state and robot's previous dialog, $a_{v,R}$.

Physical and Verbal Action Spaces. The physical and verbal action spaces, $\mathcal{A}_p$ and $\mathcal{A}_v$, are shared between both agents. Each element of these action space are a parameterized action primitive represented by the pair, $a_{p/v} = (\omega_{p/v}, \theta_{p/v})$. ω_p is the type of the physical action primitive (`open`, `pick-and-place`, etc.) and θ_p are the corresponding parameters (e.g., what object to open or pick and where to place it). We assume that humans are fully competent in executing all steps of a collaborative household manipulation task, but may be unwilling or unavailable to perform some or 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. ω_v is the type of the verbal action primitive (`ask_human_for_help`, `respond_to_human`, etc.) and θ_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 unassigned **physical** action primitives, $[a_{p,0}, ..., a_{p,K-1}]$, such as [(`pick-and-place(box, table)`, ..., `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, $G = [g_t, ..., g_{K-1}]$, 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).

IV. 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 [54], we use Q-functions to unify these two components. We assume each task step is executed by a multi-task policy π 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. III), 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.

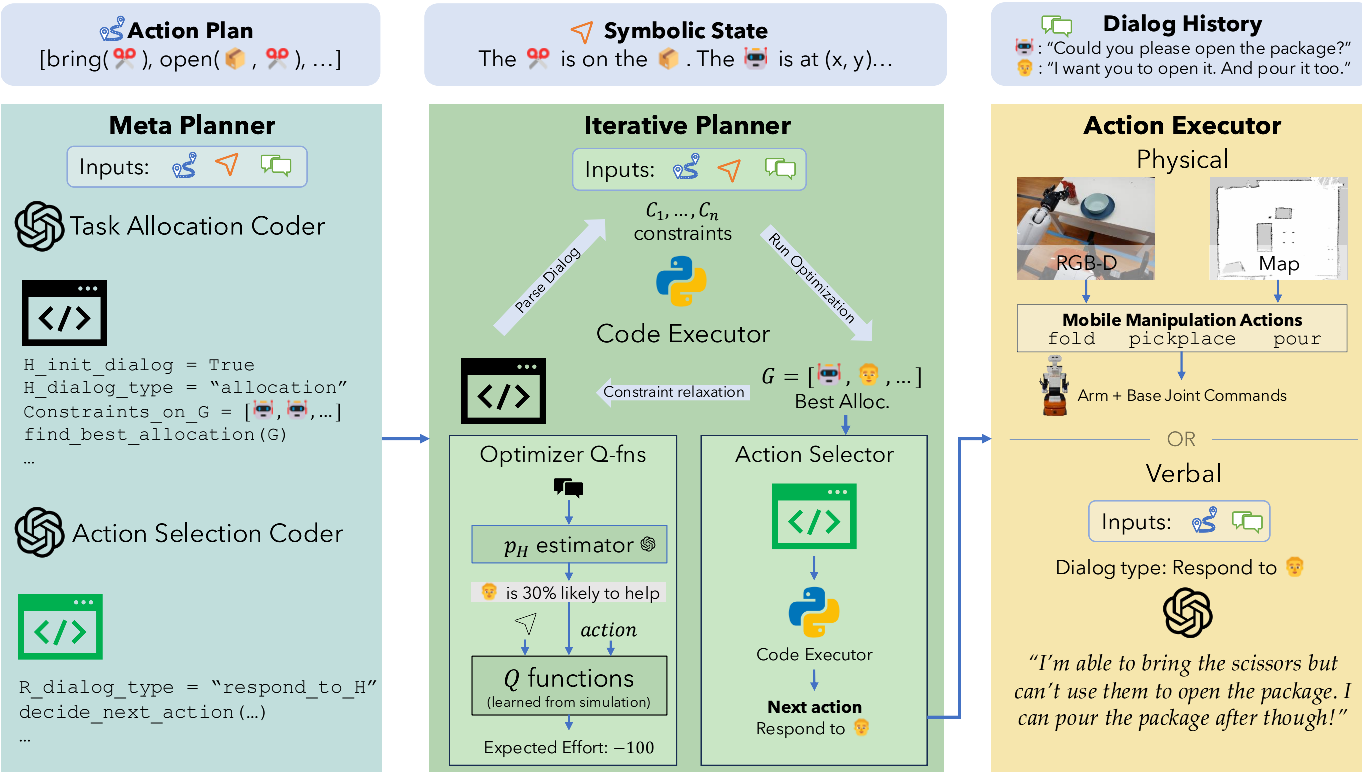


Fig. 3: MICoBot consists of 3 decision-making modules: a meta-planner that outputs a strategy for task collaboration expressed through adaptive planning code, a planner that executes the code and optimizes our objective in Equation 1 to find the next primitive action to take, and the action executor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

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 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*, α, 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, $C_1, \ldots, C_n$, 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:

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

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

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

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

L3: Action Executor. The action executor is responsible for executing the action primitive selected by the planner. For physical actions, it generates a trajectory for navigation and arm movement to reach the location and manipulate the target object while avoiding obstacles. Following a similar pipeline to Shah et al. [69], we use the `move_base` ROS package for path planning over a 2D occupancy map, and Grounding DINO [42] to segment the target object from the scene based on the natural language query in θ_t. An RGB-D camera is used to backproject segmented image pixels into a 3D point cloud, from which we identify graspable or placeable points in the robot's workspace. Inverse kinematics (IK) is then used to move the arm to these points. For verbal actions, we employ GPT-4o to generate natural language utterances to communicate with the human, based on both the intended dialog intent (e.g., help request, split proposal) and the verbal action parameters (context required for appropriate generation) from the upstream planner. Using approximately 10 in-context learning (ICL) examples, the LLM produces free-form language grounded in the task context.

Training Q-functions. MICoBot's optimization process depends on accurate approximations of the Q-functions, which capture each agent's expected effort and likelihood of success when executing a task step. To estimate the robot's Q-function (Q_R), we use the OmniGibson simulator [38], configured with a coarse model of the real-world task and environment (see Appendix for visualizations). In simulation, we execute action primitives that closely mirror those used by the physical robot, recording both completion times and failure cases. These statistics are used to construct Q_R as described earlier in this section. Since the task state is represented symbolically in both the simulator and the real-world system, the sim-to-real gap is minimal. To estimate the human's Q-function (Q_H), we assume humans do not fail at task execution. Thus, we only require time estimates for each step, which MICoBot obtains by prompting an LLM to predict how long a human would take to execute action $a_t = \omega_t(\theta_t)$, plus a travel time estimate based on the human-object distances.

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

V. Evaluation

We evaluate MICoBot in both real-world and simulated settings. In the real world, a Tiago mobile manipulator collaborates with a human user on household manipulation tasks. In simulation, we use the Mini-Behavior gridworld [32] with a simulated human, allowing for larger-scale experimentation and controlled comparisons across methods, particularly in relation to human behavior and dialog dynamics. As suggested before, a successful robotic collaborator must complete the task efficiently 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 III (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 α 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.

	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 (%, ↑)	**50**	0	**67**	0	**67**	0	**61.1**	0.0
% of task steps completed (↑)	**83**	60	**94**	29	**88**	50	**88.2**	46.4
% of steps performed by Human	21	5	60	5	35	21	38.8	10.4
% Users Preferring ... (↑)	**83**	17	**100**	0	**67**	33	**83.3**	16.7
Communicative ability (↑)	**3.3/5**	2.3/5	**4.3/5**	1.3/5	**2.8/5**	2.3/5	**3.5/5**	2.0/5
"Clearly communicated to me when it couldn't do something." (↑)	**4.3/5**	2.3/5	**3.7/5**	1.2/5	**4.2/5**	2.5/5	**4.1/5**	2.0/5
Overall Satisfaction working w/ Robot (↑)	**3.7/5**	2.7/5	**3.5/5**	1.5/5	**3.5/5**	2.5/5	**3.6/5**	2.2/5

TABLE I: 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.

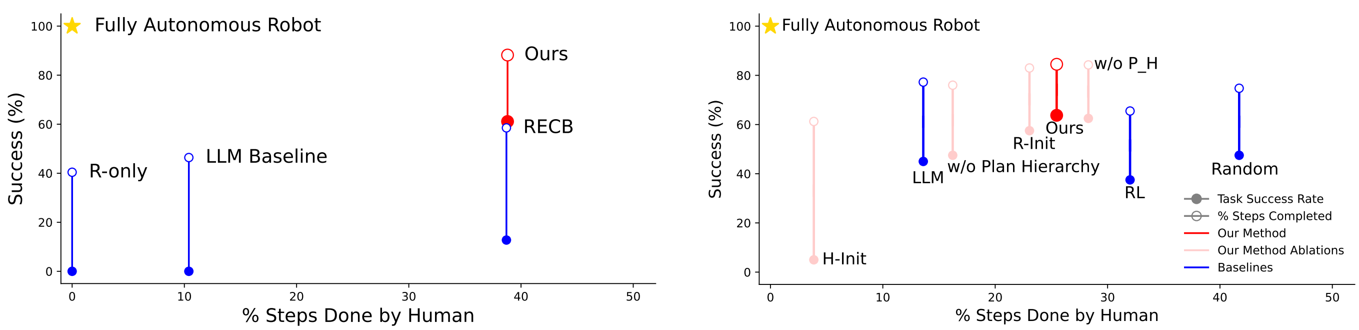


Fig. 4: In both **real-world** user studies (left) and **simulation trials** with a simulated human (right), our method (red) demonstrates the best tradeoff in achieving task success (y-axis) for a given amount of human effort (x-axis) than baselines (blue) and our method's ablations (pink).

To 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_H** (no $p_{H,t}$ estimation) and **w/o Plan Hierarchy** (where our method talks to the human in low-level ssteps).

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 I), MICoBot achieves a 61% task success rate, compared 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 I shows that 83% of users preferred our method over the LLM baseline. Among the three participants who favored the baseline, two experienced trials in which neither method completed the task, while the third preferred the baseline because it did not initiate conversation—unlike our method, which occasionally requested the user's assistance, even though the task ultimately succeeded.

Our method also significantly outperformed the baseline in overall satisfaction, scoring 3.6 out of 5.0 compared to 2.2 for the LLM. Participants rated our system as more communicative (4.1 vs. 2.0), better at initiating dialog, and more aware of its own limitations. 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.

VI. 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.

VII. 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] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as I can and not as I say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

[3] J.E. Allen, C.I. Guinn, and E. Horvtz. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications*, 14(5):14–23, 1999. doi: 10.1109/5254.796083.

[4] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru 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.

[5] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language, 2024. URL https://arxiv.org/abs/2403.01823.

[6] Adrien Bennetot, Vicky Charisi, and Natalia Díaz-Rodríguez. Should artificial agents ask for help in human-robot collaborative problem-solving?, 2020. URL https://arxiv.org/abs/2006.00882.

[7] Justin Bishop, Jaylen Burgess, Cooper Ramos, Jade B. Driggs, Tom Williams, Chad C. Tossell, Elizabeth Phillips, Tyler H. Shaw, and Ewart J. de 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.

[8] Jaime 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.

[9] Silvia Cascianelli, Gabriele Costante, Thomas A Ciarfuglia, Paolo Valigi, and Mario L Fravolini. Full-gru natural language video description for service robotics applications. *IEEE robotics and automation letters*, 3 (2):841–848, 2018.

[10] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S. Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *arXiv preprint arXiv:2209.09874*, 2022.

[11] David L. Chen, Joohyun Kim, and Raymond 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.

[12] Maximillian Chen, Xiao Yu, Weiyan Shi, Urvi Awasthi, and Zhou Yu. Controllable mixed-initiative dialogue generation through prompting. *arXiv preprint arXiv:2305.04147*, 2023.

[13] Maximillian Chen, Ruoxi Sun, Sercan O. Arık, and Tomas Pfister. Learning to clarify: Multi-turn conversations with action-based contrastive self-training, 2024. URL https://arxiv.org/abs/2406.00222.

[14] Jae-Woo Choi, Hyungmin Kim, Hyobin Ong, Youngwoo Yoon, Minsu Jang, Jaehong Kim, et al. Reactree: Hierarchical task planning with dynamic tree expansion using llm agent nodes. 2025.

[15] Jennifer 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, April 2000. Association for Computational Linguistics. doi: 10.3115/974147.974161. URL https://aclanthology.org/A00-1014/.

[16] Abhishek Das, Satwik Kottur, Jose M. F. Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[17] Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv preprint arXiv:2305.02750*, 2023.

[18] Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration, 2023. URL https://arxiv.org/abs/2305.13626.

[19] Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents, 2024. URL https://arxiv.org/abs/2311.00262.

[20] Maurizio Faccio, Irene Granata, and Riccardo 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.

[21] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An

Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

[22] Xueyang Feng, Zhi-Yuan Chen, Yujia Qin, Yankai Lin, Xu Chen, Zhiyuan Liu, and Ji-Rong Wen. Large language model-based human-agent collaboration for complex task solving, 2024. URL https://arxiv.org/abs/2402.12914.

[23] Douglas A. Few, David J. Bruemmer, and Miles 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.

[24] Qiaozi Gao, Govind Thattai, Suhaila Shakiah, Xiaofeng Gao, Shreyas Pansare, Vasu Sharma, Gaurav Sukhatme, Hangjie Shi, Bofei Yang, Desheng Zheng, Lucy Hu, Karthika Arumugam, Shui Hu, Matthew Wen, Dinakar Guthy, Cadence Chung, Rohan Khanna, Osman Ipek, Leslie Ball, Kate Bland, Heather Rocker, Yadunandana Rao, Michael Johnston, Reza Ghanadan, Arindam Mandal, Dilek Hakkani Tur, and Prem Natarajan. Alexa arena: A user-centric interactive platform for embodied ai, 2023. URL https://arxiv.org/abs/2303.01586.

[25] Prasoon Goyal, Scott Niekum, and Raymond Mooney. Using natural language for reward shaping in reinforcement learning. 2019. URL https://arxiv.org/abs/1903.02020.

[26] Prasoon Goyal, Scott Niekum, and Raymond Mooney. Pixl2r: Guiding reinforcement learning using natural language by mapping pixels to rewards. 2020. URL https://arxiv.org/abs/2007.15543.

[27] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022.

[28] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models, 2023. URL https://arxiv.org/abs/2307.05973.

[29] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea 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.

[30] Shu Jiang and Ronald 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.

[31] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv*, 2022.

[32] Emily Jin, Jiaheng Hu, Zhuoyi Huang, Ruohan Zhang, Jiajun Wu, Li Fei-Fei, and Roberto Martín-Martín. Mini-behavior: A procedurally generated benchmark for long-horizon decision-making in embodied ai. *arXiv preprint 2310.01824*, 2023.

[33] Siddharth Karamcheti, Megha Srivastava, Percy Liang, and Dorsa Sadigh. Lila: Language-informed latent actions. In *5th Annual Conference on Robot Learning*, 2021. URL https://arxiv.org/pdf/2111.03205.

[34] Ross A. Knepper, Todd Layton, John Romanishin, and Daniela 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.

[35] Noriyuki Kojima, Alane Suhr, and Yoav Artzi. Continual learning for grounded instruction generation by observing human following behavior, 2021. URL https://arxiv.org/abs/2108.04812.

[36] Benjamin Kolb, Leon Lang, Henning Bartsch, Arwin Gansekoele, Raymond Koopmanschap, Leonardo Romor, David Speck, Mathijs Mul, and Elia Bruni. Learning to request guidance in emergent communication, 2019. URL https://arxiv.org/abs/1912.05525.

[37] Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1321. URL https://aclanthology.org/D17-1321/.

[38] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, Karen Liu, Jiajun Wu, and Li 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.

[39] Chengshu Li, Jacky Liang, Andy Zeng, Xinyun Chen, Karol Hausman, Dorsa Sadigh, Sergey Levine, Li Fei-Fei, Fei Xia, and Brian Ichter. Chain of code: Reasoning with a language model-augmented code emulator. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix 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.

[40] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *arXiv preprint arXiv:2209.07753*, 2022.

[41] Toru Lin, Minyoung Huh, Chris Stauffer, Ser-Nam Lim, and Phillip Isola. Learning to ground multi-agent communication with autoencoders, 2021. URL https://arxiv.org/abs/2110.15349.

[42] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2023.

[43] Kenzo Lobos-Tsunekawa, Akshay Srinivasan, and Michael Spranger. Ma-dreamer: Coordination and communication through shared imagination, 2022. URL https://arxiv.org/abs/2204.04687.

[44] Ryan Lowe, Abhinav Gupta, Jakob Foerster, Douwe Kiela, and Joelle Pineau. On the interaction between supervision and self-play in emergent communication, 2020. URL https://arxiv.org/abs/2002.01093.

[45] Xusheng Luo, Shaojun Xu, and Changliu Liu. Obtaining hierarchy from human instructions: an llms-based approach. In *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.

[46] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *Robotics: Science and Systems*, 2021. URL https://arxiv.org/abs/2005.07648.

[47] Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*, 2023.

[48] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models, 2024. URL https://arxiv.org/abs/2310.12931.

[49] Yecheng Jason Ma, William Liang, Hungju Wang, Sam Wang, Yuke Zhu, Linxi Fan, Osbert Bastani, and Dinesh Jayaraman. Dreureka: Language model guided sim-to-real transfer. In *Robotics: Science and Systems (RSS)*, 2024.

[50] Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models, 2023. URL https://arxiv.org/abs/2307.04738.

[51] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2021. URL https://arxiv.org/abs/2112.03227.

[52] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned imitation learning. *arXiv preprint arXiv:2204.06252*, 2022.

[53] Bilge Mutlu, Jodi Forlizzi, and Jessica 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.

[54] Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making, 2025. URL https://arxiv.org/abs/2406.17098.

[55] Suraj Nair, Eric Mitchell, Kevin Chen, Brian Ichter, Silvio Savarese, and Chelsea 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.

[56] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[57] Manisha Natarajan, Chunyue Xue, Sanne van Waveren, Karen Feigh, and Matthew Gombolay. Mixed-initiative human-robot teaming under suboptimality with online bayesian adaptation, 2024. URL https://arxiv.org/abs/2403.16178.

[58] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[59] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025, 2022.

[60] Rohan Paleja, Michael Munje, Kimberlee Chang, Reed Jensen, and Matthew Gombolay. Designs for enabling collaboration in human-machine teaming via interactive and explainable systems, 2024. URL https://arxiv.org/abs/2406.05003.

[61] Jannik Peters, Constantin Waubert de Puiseau, Hasan Tercan, Arya Gopikrishnan, Gustavo Adolpho Lucas de Carvalho, Christian Bitter, and Tobias Meisen. Emergent language: a survey and taxonomy. *Autonomous Agents and Multi-Agent Systems*, 39(1), March 2025. ISSN 1573-7454. doi: 10.1007/s10458-025-09691-y. URL http://dx.doi.org/10.1007/s10458-025-09691-y.

[62] Kun Qian, Ahmad Beirami, Satwik Kottur, Shahin Shayandeh, Paul Crook, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. Database search results disambiguation for task-oriented dialog systems, 2021. URL https://arxiv.org/abs/2112.08351.

[63] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[64] Shreyas Sundara Raman, Vanya Cohen, David Paulius, Ifrah Idrees, Eric Rosen, Ray Mooney, and Stefanie Tellex. Cape: Corrective actions from precondition errors using large language models. In *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[65] Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners, 2023. URL https://arxiv.org/abs/2307.01928.

[66] Andres Rosero, Faustina Dinh, Ewart J. de Visser, Tyler Shaw, and Elizabeth Phillips. Two many cooks: Understanding dynamic human-agent team communication and perception using overcooked 2, 2021. URL https://arxiv.org/abs/2110.03071.

[67] Mario Selvaggio, Marco Cognetti, Stefanos Nikolaidis, Serena Ivaldi, and Bruno Siciliano. Autonomy in physical human-robot interaction: A brief survey. *IEEE Robotics and Automation Letters*, 6(4):7989–7996, 2021.

[68] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. *arXiv preprint arXiv:2309.14320*, 2023.

[69] Rutav Shah, Albert Yu, Yifeng Zhu, Yuke Zhu, and Roberto Martín-Martín. Bumble: Unifying reasoning and acting with vision-language models for building-wide mobile manipulation. 2025.

[70] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[71] Afagh Mehri Shervedani, Siyu Li, Natawut Monaikul, Bahareh Abbasi, Barbara Di Eugenio, and Milos Zefran. Multimodal reinforcement learning for robots collaborating with humans, 2024. URL https://arxiv.org/abs/2303.07265.

[72] Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z. Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at your robot: Improving on-the-fly from language corrections, 2024. URL https://arxiv.org/abs/2403.12910.

[73] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. URL https://arxiv.org/abs/2502.19417.

[74] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[75] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *Conference on Robot Learning*, 2022.

[76] Andrew Silva, Nina Moorman, William Silva, Zulfiqar Zaidi, Nakul Gopalan, and Matthew Gombolay. Lanconlearn: Learning with language to enable generalization in multi-task manipulation. In *IEEE Robotics and Automation Letters*, 2021.

[77] Sukriti Singh, Anusha Srikanthan, Vivek Mallampati, and Harish Ravichandar. Concurrent constrained optimization of unknown rewards for multi-robot task allocation, 2023. URL https://arxiv.org/abs/2305.15288.

[78] Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multitask reinforcement learning with context-based representations. *arXiv preprint arXiv:2102.06177*, 2021.

[79] Alane Suhr, Claudia Yan, Charlotte Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. Executing instructions in situated collaborative interactions, 2022. URL https://arxiv.org/abs/1910.03655.

[80] DeepMind Interactive Agents Team, Josh Abramson, Arun Ahuja, Arthur Brussee, Federico Carnevale, Mary Cassin, Felix Fischer, Petko Georgiev, Alex Goldin, Mansi Gupta, Tim Harley, Felix Hill, Peter C Humphreys, Alden Hung, Jessica Landon, Timothy Lillicrap, Hamza Merzic, Alistair Muldal, Adam Santoro, Guy Scully, Tamara von Glehn, Greg Wayne, Nathaniel Wong, Chen Yan, and Rui Zhu. Creating multimodal interactive agents with imitation and self-supervised learning, 2022. URL https://arxiv.org/abs/2112.03763.

[81] Shivam Vats, Oliver Kroemer, and Maxim Likhachev. Synergistic scheduling of learning and allocation of tasks in human-robot teams, 2022. URL https://arxiv.org/abs/2203.07478.

[82] Manuela Veloso, Joydeep Biswas, Brian Coltin, and Stephanie 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.

[83] Huaxiaoyue Wang, Kushal Kedia, Juntao Ren, Rahma Abdullah, Atiksh Bhardwaj, Angela Chao, Kelly Y Chen, Nathaniel Chin, Prithwish Dan, Xinyi Fan, Gonzalo Gonzalez-Pumariega, Aditya Kompella, Maximus Adrian Pace, Yash Sharma, Xiangwan Sun, Neha Sunkara, and Sanjiban Choudhury. Mosaic: A modular system for assistive and interactive cooking, 2024. URL https://arxiv.org/abs/2402.18796.

[84] Mark Woodward, Chelsea Finn, and Karol Hausman. Learning to interactively learn and assist, 2019. URL https://arxiv.org/abs/1906.10187.

[85] Albert Yu and Raymond J Mooney. Using both demonstrations and language instructions to efficiently learn robotic tasks. *arXiv preprint arXiv:2210.04476*, 2022.

[86] Tian Yu, Jing Huang, and Qing 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.

[87] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. Language to rewards for robotic skill synthesis, 2023. URL https://arxiv.org/abs/2306.08647.

[88] Xiao Yu, Maximillian Chen, and Zhou Yu. Prompt-based monte-carlo tree search for goal-oriented dialogue policy planning, 2023. URL https://arxiv.org/abs/2305.13660.

[89] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *CoRR*, abs/2111.07991, 2021. URL https://arxiv.org/abs/2111.07991.

[90] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023.

Appendix

A. Real-world Task Descriptions

1) Task Plans: Fig. 5 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.

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)`

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

1) Bring parts to coffee table.
   a) `pickplace(parts_tray, coffee_table)`
   b) `pickplace(wheels, coffee_table)`
2) Assemble wheels.
   a) `pickplace(drill, coffee_table)`
   b) `put_on(wheels, car, drill)`

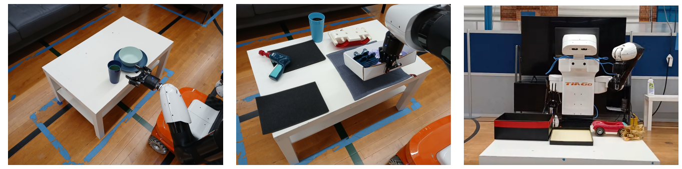


Fig. 5: Real-world tasks from left to right: pouring package into bowl, assembling toy car, and packing gift box.

3) Switch drill bit.
   a) `pickplace(hex_drill_bit, coffee_table)`
   b) `switch(hex_drill_bit, drill)`
4) Assemble rest of car.
   a) `put_on(window, car, drill)`
   b) `put_on(seats, car, drill)`

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

1) Assemble box.
   a) `fold(box_flap)`
2) Put in gift.
   a) `pickplace(gift_tissue_paper, box)`
   b) `pickplace(toy_car, box)`
3) Seal the box.
   a) `cover(box_lid, box)`
   b) `pickplace(ribbons, coffee_table)`
   c) `wrap(ribbons, box)`
4) Decorate the box.
   a) `cut_put(tape, scissors, box)`
   b) `pickplace(gift_bow, box_lid)`

B. Dialog Excerpts from our User Studies

1) Convincing a Reluctant User to Perform a Step: Our system is able to flexibly handle long dialog interactions like this one on Task 1 (opening and pouring the package into the bowl), where it manages to convince the human to help with opening the package.

Robot *"Could you please open the package using the scissors on the coffee table? Thank you so much!"*

Human *"There are no scissors on the coffee table."*

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

Human *"I think you should bring the scissors, and then you should use the scissors to open the package."*

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

Human *"Ok, if you bring the scissors, then I will open the package with the scissors."*

Robot *"Great! I'll happily place the scissors on the coffee table. Thank you for taking care of opening the package with the scissors!"*

Robot [brings the scissors]
Human *"Thanks for bringing the scissors!"*
Robot *"Could you please open the package with scissors and place it on the coffee table? Thank you!"*
Human *"Ok, I will do that now!"*
Human [opens the package]

2) Failure Case: This is an example of MICoBot unable to act on the motion-level feedback (making the tissue paper "lay flatter") that the human wants the robot to perform. Additionally, MICoBot misunderstands the human and thinks they are talking about the ribbons step when the human is still talking about the tissue paper.

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

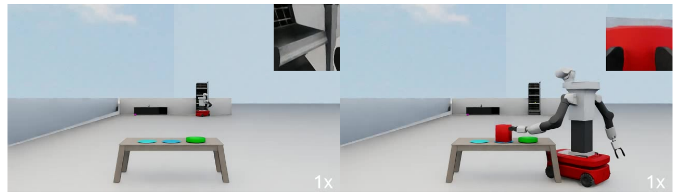
For the complete set of dialogs and operations in our human trials, please visit our website at https://mico-bot.github.io/

C. MICoBot Implementation Details

1) Robot Q-function Q_R training in OmniGibson: To train Q-functions for the robot, we first create a simulated OmniGibson environment with a PAL Tiago robot and an environment that roughly matches the relative locations of the relevant furnitures and objects. We then implemented each real-world skill first in OmniGibson. Fig. 6 depicts example frames from primitives in task 1 and task 3 we ran in the OmniGibson simulator to collect sample Q-values for each skill.

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

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 ℓ_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.



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).

Fig. 6: 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.

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.

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.

D. Detailed Simulation Results

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 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.

2) Simulation Experiments: In Table II, 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 II are plotted in Fig. 4.

E. User Study Details

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

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

3) Failure Analysis: In the real world, the main sources of error of our method were as follows:

1) Task 1: Cut and Pour Package into Bowl. 3 failed trials out of 6.
   - 2 primitive errors (pouring missed the bowl, package grasping pressed into the shelf and wasn't placed properly)
   - 1 perception error
2) Task 2: Assemble Toy Car. 2 failed trials out of 6. 1 failure that was rectified by human.

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

Method	Metric	Human Parameters (Mood, $\tilde{p}_{H,t}$)								Avg. (%)
		Positive Mood				Negative Mood				
		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

- 1 dialog error: user wanted robot to perform a non-step plan outside of its capabilities, and refused when robot said it wasn't able to perform it
- 1 primitive error: robot did not release its grasp of the drill.
- 1 metaplanner dialog parsing error

3) Task 3: Pack Gift Box. 2 failed trials out of 6.
   - 1 primitive error: placed bow on box but bow dropped to floor as gripper retracted
   - 1 termination condition triggered: rejected robot's help request proposal 3 times in a row.

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

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

4) Fault Recovery: Sometimes, the code produced by the metaplanner is not executable. For fault recovery, the metaplanner is automatically re-queried up to 2 additional times to create code. If these attempts also produce non-executable code, the most recent dialog from the human is ignored for 2 further, automated metaplanner requeries. These re-queries are handled by a try-except block in the iterative planner module of MICoBot.

F. RL Baseline Details

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

We initially trained the RL policy on two simulated human settings: one where the human ground 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 non-zero training returns after thousands of iterations on the latter setting, so we only reported results on

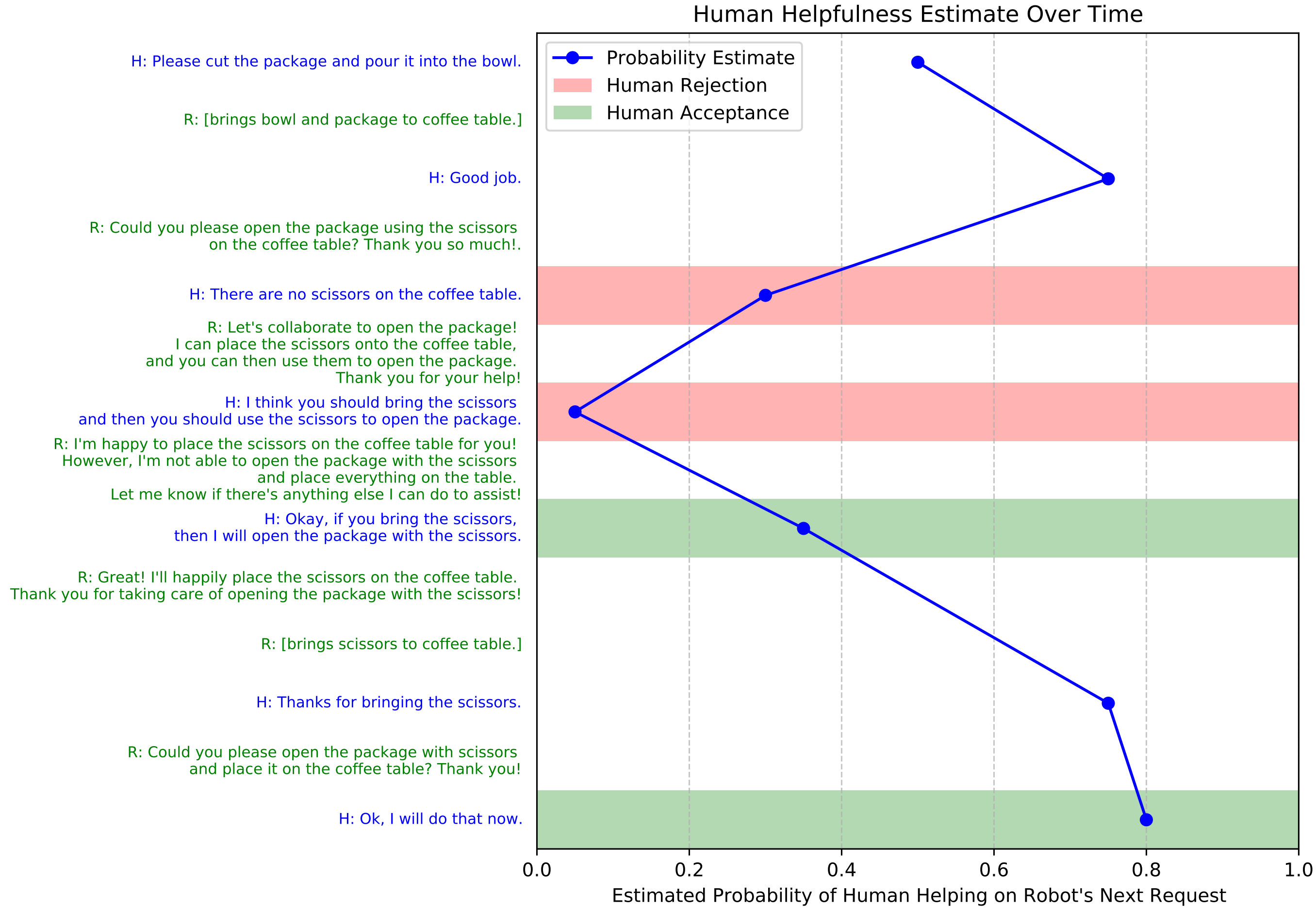


Fig. 7: From a real-world user study: MICoBot's $p_{H,t}$ estimation (blue line) reacts in real time to the human's rejections (red), acceptances (green), and encouraging remarks. All dialog is shown as y-labels. Green text denotes robot actions/dialog, and blue text denotes human dialog. The timestep t increases from top to bottom on the y-axis.

the former setting, which explains why the RL policy does not perform well when $\tilde{p}_{H,t}$ is low.

G. Additional Experimental Investigations

In addition to those discussed in Section V, we explore the following additional experimental questions.

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

First, we examine in Fig. 7 a real-world instance of how well MICoBot can estimate the probability of the human helping on the next turn during the course of a user study. After the robot's help request was rejected twice in a row (top 2 red horizontal stripes), the robot's helpfulness estimate of the human plummets to 0.05. However, after the robot explains its incapacity to use scissors, the human accepts the next two help requests (in green) and the robot's helpfulness estimate of the 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 III 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 III 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

TABLE III: 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)

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 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.

H. Further Connections to Prior Work

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 [16, 71] or IL (imitation learning) [59, 80]. Research on language emergence in multiagent systems [61, 37] has also examined how cooperative agents learn to communicate through latent representations or natural language when performing simulated robotic tasks [41, 44, 84, 43, 36]. 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.

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 [29, 46, 51, 52, 70, 78, 76, 33, 31, 68, 85], LLMs as task planners [27, 2, 10, 64], code-based policies [40, 39, 28], hierarchical policies [73, 72, 5] and planners [14, 45], vision-language representations [63, 89, 90] for robotic control [56, 74, 75], and language-based reward shaping for RL policies [55, 25, 26, 21, 47–49, 87].