IEEE Transactions on Robotics (T-RO) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Shared Autonomy of a Robotic Manipulator for Grasping under Human Intent Uncertainty using POMDPs

J-Anne Yow, Neha Priyadarshini Garg and Wei Tech Ang

***Abstract*—In shared autonomy (SA), accurate user intent prediction is crucial for good robot assistance and avoiding user-robot conflicts. Prior works have relied on passive observation of joystick inputs to predict user intent, which works when the goals are clearly separated or when a common policy exists for multiple goals. However, they may not work well when grasping objects to perform daily activities, as there are multiple ways to grasp the same object. We demonstrate the need for active information-gathering in such cases and show how this can be done in a principled manner by formulating SA as a discrete action Partially Observable Markov Decision Process (POMDP), reasoning over high-level actions. One of our insights is that apart from having explicit information-gathering actions and goal-oriented actions, it is important to have actions that move towards a distribution of goals and provide no assistance in the POMDP action space. Compared to a method with no active information-gathering, our method performs tasks faster, requires less user input, and decreases opposing actions, especially for more complex objects, getting higher ratings and preference in our user study.**

***Index Terms*—Physical human-robot interaction, human performance augmentation, physically assistive devices, planning under uncertainty**

I. Introduction

With an ageing population and rising prevalence of stroke in young adults [1], there is an increasing number of people who require assistance to perform activities of daily living (ADLs). Coupled with the shortage of caregivers globally [2], robotic assistive devices such as robotic manipulators show promise in assisting those with motor impairments to perform ADLs, such as grasping objects, eating and dressing. However, controlling a robotic arm is challenging due to the difference in the degrees of freedom (DoF) between robotic manipulators and their control interfaces.

Commercially available control interfaces such as joysticks, sip-and-puff devices and head arrays are commonly 1-DoF to 3-DoF, while a robotic arm is 7-DoF (6-DoF for translation and rotation and 1-DOF for opening/closing the gripper). Thus, users have to switch control modes to control all DoF of the arm. This increases the cognitive burden on the user and adversely affects task performance [3].

Shared autonomy combines robot inputs and human inputs, allowing both robot and human to interact continuously to achieve a common task [4]. The robot may assist the user by sharing the control burden, which is helpful when controlling a 7-DoF robotic manipulator using low-DoF control interfaces. In this work, we focus on the control of a 7-DoF robot using a 3-DoF joystick through shared autonomy. While controlling a robot arm with a 3-DoF joystick is admittedly easier than a 1-DoF joystick, it is still challenging due to the need to switch between control modes, and controlling the orientation of a robot arm may not come intuitively to most users [5].

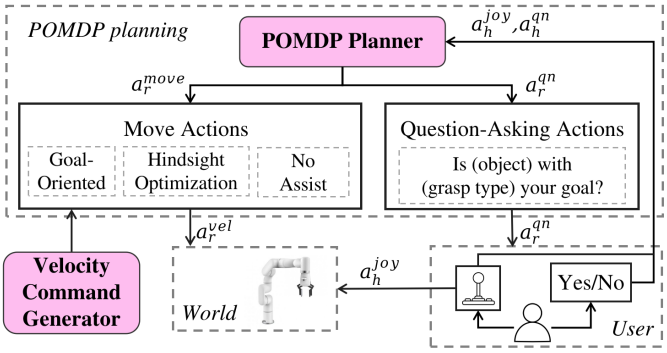


Fig. 1. Overview of our shared autonomy framework. The POMDP model plans over high-level actions a_r, either moving towards a goal or asking questions about a goal. The low-level actions executed by the robot a_r^{vel} for move actions are generated by a velocity command generator for each goal and for a distribution of goals using hindsight optimization [6].

State-of-the-art shared autonomy methods for robotic manipulators [6]–[14] predict the human intention or goal by passively observing the user's joystick inputs while providing assistance. To avoid assisting to the wrong goal, shared autonomy algorithms either assist only when the confidence in the predicted goal is high [10] or assist towards a distribution of goals [6]. In both methods, the underlying assumption is that passive observation of human inputs while assisting is sufficient to predict human intention. This assumption works well when the goals are well separated from each other or when there is a common policy for the goals.

However, real-world scenarios are generally cluttered, and we usually grasp the objects in our daily lives differently based on the task. Existing works [6], [10]–[14] primarily consider objects as possible user goals without considering how to grasp the object. These works either fix the grasp pose[1] for each object or select the grasp pose that is closest to the gripper, without further consideration of the grasping strategy for a given object. This simplifies the problem as the size of the set of possible goals becomes small and a common policy is applicable to multiple goals. However, the user might not want

This work is supported by the Singapore National Robotics Programme, Robotics Enabling Capabilities and Technologies (RECT) Grant 1822500095 and the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. *(Corresponding author: J-Anne Yow)*

The authors are with the Rehabilitation Research Institute of Singapore (RRIS), Clinical Science Building, 308232 Singapore. RRIS is a joint research institute by Nanyang Technological University (NTU), Agency for Science, Technology and Research (A*STAR) and National Healthcare Group (NHG), Singapore. (e-mail: {janne.yow, np.garg and WTAng}@ntu.edu.sg)

[1]A grasp pose is a 6-DoF pose of the robot gripper in which closing the gripper would grasp the object

to grasp the object with the closest or pre-defined grasp pose, which may cause the user to fight with the system.

To address this issue, we redefine "goals" such that our "goals" consider different possible ways to grasp the same object. Thus, our "goals" include the object and grasp type. Since each object may have multiple goals due to different ways to grasp (e.g. from the top or side), the goals are no longer well separated. Predicting intent accurately only from passive observation of human inputs is very difficult in such scenarios [15], which we observe in our experiments. To provide meaningful assistance in such scenarios, the system should be able to explore, i.e. actively gather information from the human when uncertainty about human intention is high, and exploit, i.e. move towards the goal when uncertainty about human intention is low.

In this work, we define active information-gathering actions as verbal questions since language is the most intuitive and effective form of communication. Unfortunately, active information gathering would increase task completion time, and querying the user too much could be annoying to the user. Thus, it is important for the system to reason *whether* to ask questions, *when* to ask questions, and *what* questions to ask in a shared autonomy setting.

Partially Observable Markov Decision Processes (POMDPs) [16] provide a principled framework for balancing exploration and exploitation. They have been used for active information gathering in human-robot adaptation and collaboration [13], [17], [18]. While prior work has formalized shared autonomy as a continuous action POMDP [6], the QMDP (POMDP with Q-values) [19] or hindsight optimization approximation was used to solve the model since solving a POMDP with continuous actions is intractable. With the QMDP approximation, there is no active information gathering. While this approximation works well in many scenarios, the lack of information gathering could lead to a convergence to the wrong goal in complex scenarios where the goals are close to each other.

Thus, we formulate the problem of shared autonomy as a discrete action POMDP, introducing active information-gathering actions by performing decision-making on high-level actions (Fig. 1) instead of continuous low-level actions, such as moving left or right. Determining appropriate high-level actions in shared autonomy is not as straightforward as in human-robot collaboration, since high-level actions are not inherent to the task. We carefully define our high-level move actions to include *goal-oriented* actions that move towards each goal, a *hindsight optimization* action that assists towards a distribution of goals, and a *no assist* action that allows the user to take full control resulting in implicit information gathering. Using discrete high-level actions makes our POMDP solvable by state-of-the-art POMDP solvers [20]–[24] and allows active information gathering, which leads to better assistance in daily living scenarios involving objects with multiple ways to grasp.

Overall, the contributions of this paper include 1) demonstrating the need for active information gathering in shared autonomy 2) modelling shared autonomy as a discrete-action POMDP by defining relevant high-level actions, which includes question-asking actions for explicit information gathering, a *no assist* action to allow arbitration of control, and a *hindsight optimization* action to assist towards a distribution of goals, and 3) two user studies consisting of 18 subjects (3 with upper limb impairments) and 15 subjects respectively, comparing our method against hindsight optimization [6] and manual control. The results of the user studies show that our method performed at least as well as hindsight optimization for simple tasks, and outperformed hindsight optimization in scenarios where information-gathering is required. In the second user study, we also show that our method can scale to scenes with a larger number of goals. The user surveys showed greater preference for our method compared to hindsight optimization, citing greater control, a better understanding of the robot assistance, and less effort required to perform the task.

II. Related Work

A. Shared Autonomy

Shared autonomy can be broadly divided into planning-based approaches [3], [6]–[14] and learning-based approaches [25]–[27]. Learning-based algorithms learn a shared autonomy policy from the control input data collected through user demonstrations. Such approaches require substantial data, which is difficult to acquire for shared autonomy tasks. Thus, we focus on planning-based algorithms in this work. Planning-based approaches compute the robot's control action by minimizing a cost function. The cost function is usually defined such that the robot moves towards the user's intended goal. Planning-based approaches can be further classified based on how the assistance is provided or how the human and robot actions are blended together.

1) Assistance Based Classification: Planning-based approaches can be divided into three categories based on how they provide assistance – 1) predict-then-act, 2) assist under uncertainty without information gathering and 3) assist under uncertainty with information gathering.

Predict-then-Act approaches [8]–[10] move to the most likely goal predicted based on the user's joystick input when the confidence in prediction is sufficiently high. The main drawback is that there is no assistance for a significant portion of the task when the prediction confidence is low.

Assist Under Uncertainty without Information Gathering approaches [6] address the drawbacks of predict-then-act approaches by assisting towards a distribution of goals, or belief, even when the confidence of the prediction is low for the most likely goal. This approach works well when a common policy exists for different goals. However, the assistance provided can be wrong when the belief starts to converge on a wrong goal. As these approaches rely solely on passive observation of the user's inputs, the user may need to exert significant effort to oppose the assistance provided to correctly update the system's belief.

Assist Under Uncertainty with Information Gathering approaches address this issue by actively seeking information from the user to update the belief correctly. Active information gathering for shared autonomy has been studied in [14], where implicit information gathering actions are taken when

the entropy of belief is high. By reducing the entropy of belief, the system had a faster belief convergence. However, the assistance provided did not improve. We observe that a reduction in entropy is not always a good indicator of a better policy, as a good policy can be found using hindsight optimization [6] even when the entropy is high. Thus, overall improvement in assistance is not achieved in [14]. In our work, we use POMDPs to reason whether information-gathering actions will lead to better assistance after reducing belief entropy, performing information-gathering actions only when needed.

POMDPs have also been used to assist with active information gathering under uncertainty in human adaptability [13] for a grasping task. An assumption is made that the robot knows the optimal goal. This simplifies the action space and changes the problem from shared autonomy to human adaptability as there is no uncertainty about the goal. This assumption cannot be made in real-world scenarios for a grasping task as the optimal goal is either difficult to define or unknown.

To the best of our knowledge, we are the first to demonstrate that active information gathering can be performed in shared autonomy by modelling the problem as a discrete action POMDP, instead of applying a QMDP (POMDP with Q-values) approximation.

2) Arbitration Based Classification: Another way of classifying the planning-based shared autonomy approaches is based on how the human and robot actions are blended together. They can be blended in 2 ways 1) outer-loop blending and 2) inner-loop blending.

In *outer-loop blending* [8]–[10], [12], [13], an optimal robot action a_r is either assumed to be known [13] or computed based on most likely goal [8]–[10], [12]. The final action a_{total} is computed by blending the user input a_h and the optimal robot action a_r based on an arbitration factor α i.e. $a_{total} = (1-\alpha)a_h + \alpha a_r$. These approaches typically use a predict-then-act paradigm, where α is determined based on the confidence in the user intent prediction. POMDPs have also been used to determine the arbitration factor by predicting the human adaptability [13] for a grasping task. The observations of the human action affect the prediction of the human adaptability, which determines the arbitration – if the human is non-adaptable, the robot follows the human; if the human is adaptable, the robot insists on its policy.

The main advantage of outer-loop blending approaches is that the assistance is provided only when the robot is confident of the human's goal. However, this also leads to cases where no assistance is provided for a significant portion of the task. Such approaches also do not consider the human action when computing the optimal robot action, which could lead to the rejection of sub-optimal robot actions that could be optimal when combined with human actions.

Thus, in this work, we use an inner-loop blending approach. *Inner-loop blending* approaches [6], [14] take into account both the human action and various potential robot actions when determining the optimal action that minimizes a cost function for a given state. The human and robot actions are usually blended together using a fixed arbitration factor α inside the cost function. These approaches provide assistance even when the prediction confidence is low since the arbitration factor is fixed. However, there is a higher chance of providing the wrong assistance. In our work, we include explicit (via asking questions) and implicit (via no assistance) information-gathering actions, providing the system with an additional option of gathering more information if all the other goal-oriented actions have a higher cost. By modelling this problem as a discrete action POMDP, we ensure that additional information is gathered only when needed, leading to better assistance.

B. Robots Querying People

Natural language has been applied in robotics for communication between humans and robots since language is a natural way to convey information to humans. Prior works have used language to query humans to get help when the robot encounters problems [28], [29] and to disambiguate uncertainties [17], [18], [30]–[32].

POMDPs have been used to disambiguate uncertainties in language expressions or visual perception for a robot grasping task [30]–[32]. Given a language instruction (e.g. "Pick up the cup."), the system can decide whether to perform an information-gathering action in the form of asking an object-specific question (e.g. "Do you mean the *blue* cup?", "Do you mean the *leftmost* cup?"), or to pick up the desired object. Users can provide "yes" or "no" verbal responses, or provide additional descriptions, e.g. "No, the *red* one", which are then added to the instruction and visually grounded. In such settings, the human intent is known, and humans are used as a source of information to reduce uncertainty about the environment. Humans do not share the same action space with robots, i.e. humans do not have direct control over the low-level gripper movements, so deciding when to ask questions is relatively easier compared to our setting as the robot does not need to consider the human action during decision making.

In human-robot collaboration, POMDPs have been used to decide what type of verbal communication the robot should engage in [17], [18], such as whether to *inform* users about the robot state, *command* users to perform an action, or *ask* users about the human state. Similarly, humans and robots do not have the same action space in these settings.

While not robots querying humans, verbal communication has also been incorporated to disambiguate between different latent actions of tasks in shared autonomy [27], [33]. However, the burden of communication is placed on the human, thus no planning is required by the robot to reason when and what to communicate. In our work, the burden of communication is placed on the robot, so the robot needs to decide what questions to ask and when to ask questions.

III. Problem Statement and Model Specification

Given a user controlling a robotic arm with a joystick to grasp a desired object with a desired grasp pose, we aim to generate robot actions to assist the user in reaching the desired grasp pose with minimal joystick control inputs and time. The robot can observe the objects placed on a table from an RGBD image of the scene. However, the robot does not know the

user's intended object and grasp pose beforehand. It can be inferred implicitly from observing the user's joystick inputs, or explicitly by asking questions from the user.

To generate the best robot action under human intent uncertainty, we model this problem as a POMDP, which is defined by the tuple $\langle \mathcal{S}, \mathcal{A}_r, \mathcal{Z}, \mathcal{T}, \mathcal{O}, \mathcal{R}, \gamma \rangle$. To maintain tractability, we model shared autonomy as a discrete-action POMDP, defining relevant high-level actions such as *no assist* and *hindsight optimization* actions for better performance. Furthermore, goals are defined at two levels, allowing goals to be refined online. In the following sections, we describe the POMDP model.

A. State Space $\mathcal{S}$

$\mathcal{S}$ is the set of states that the robot could be in. We represent state $s \in \mathcal{S}$ as a tuple $\langle x, g \rangle$, where $x \subseteq \mathbb{R}^6$ is the 6-DoF pose of the robot gripper, $g \in \mathcal{G}$ is the user's goal and $\mathcal{G}$ is the set of possible user's goals. The gripper pose x is fully observed and the user's goal g is hidden.

1) Goal Definition: To account for different ways to grasp an object, we consider the object (o) and grasp type (θ_o) in our goal, which we term a high-level target. Each high-level target has a list of grasp poses ($\mathcal{K}_{\theta_o}$), which we term a low-level target. Thus, the user's goal g is represented as a tuple $\langle o, \theta_o, \mathcal{K}_{\theta_o} \rangle$.

The high-level target contains the object to grasp $o \in O$ and the grasp type $\theta_o \in \Theta_o$ for that object o. The set of objects O is determined from an RGBD image of the scene using Mask R-CNN [34]. The set of grasp types Θ_o can have different values depending on the object's intended use. In this work, we categorise the grasp types based on the gripper orientation, such as a top or side grasp. Thus, size of the set of goals is $|\mathcal{G}| = |O| * |\Theta_o|$.

However, for the same object and grasp type, there could still be multiple grasp poses, e.g. grasping a cup with a top grasp at the left or right rim. Thus, the low-level target contains the list of grasp poses $\mathcal{K}_{\theta_o}$ for the object o and grasp type θ_o. The grasp pose $k_c \in \mathcal{K}_{\theta_o}$ that is closest to the robot's gripper pose x in terms of translational and rotational distance represents the user's desired grasp pose for the goal g. To allow our model to reason over different grasp poses when the high-level target is known, i.e. when the user answers "yes" to a question asked, we redistribute the grasp poses in $\mathcal{K}_{\theta_o}$ for highly likely high-level targets (See Section III-G).

Grasp Pose Detection (GPD) [35] is used to generate the set of grasp poses $\mathcal{K}_o$ from a masked point cloud of the object o. We then cluster them into different grasp types based on their grasp orientation to obtain the list of grasp poses $\mathcal{K}_{\theta_o}$ for each grasp type.

B. Action Space $\mathcal{A}_r$ and Transition Function $\mathcal{T}$

$\mathcal{A}_r$ is the set of high-level actions that the robot can take. There are two types of actions: move actions ${\mathcal{A}_r}^{move}$, and question-asking actions ${\mathcal{A}_r}^{qn}$.

Move actions consist of velocity commands for the gripper in the Cartesian space ${a_r}^{move} \in {\mathcal{A}_r}^{move} \subseteq \mathbb{R}^6$. Theoretically, the size of $|{\mathcal{A}_r}^{move}|$ can be infinite. However, most of the velocity commands would lead to sub-optimal solutions. We reduce the size of the action space ${\mathcal{A}_r}^{move}$ by considering only a few high-level actions that either move the robot towards the goals or follow the user action. In addition to the obvious goal-oriented actions, we define two other move actions: *hindsight optimization* action $a_{r_{HO}}^{move}$ and *no assist* action $a_{r_{None}}^{move}$.

The *hindsight optimization* action $a_{r_{HO}}^{move}$ allows the model to choose an action for moving towards a distribution of goals, which is important at the start when all the goals are equally likely. This prevents moving towards a random goal through goal-oriented actions or asking questions about random goals through question-asking actions, which can be annoying for the user.

The *no assist* action $a_{r_{None}}^{move}$ simply executes the velocity command corresponding to the user's joystick input. This action allows the robot to arbitrate control and gives the user full control when all other move actions oppose the user action, or when uncertainty is high.

Thus, ${\mathcal{A}_r}^{move}$ consists of *goal-oriented* actions (a_g^{move} for each goal g), a *hindsight optimization* action $a_{r_{HO}}^{move}$ and a *no assist* action $a_{r_{None}}^{move}$. The size of move actions is $|{\mathcal{A}_r}^{move}| = |\mathcal{G}| + 2$.

A velocity command generator is used to convert the high-level move actions ${a_r}^{move}$ to robot velocity commands a_r^{vel}. Each goal-oriented action a_g^{move} moves the gripper pose x in the direction of the goal g. If k_c is the closest grasp pose to the current gripper pose x among the list of grasp poses in goal g, then $a_g^{vel} \propto (k_c - x)$. For the *hindsight optimization* action $a_{r_{HO}}^{move}$, the robot velocity command is computed using the algorithm in [6] that assists towards a distribution of goals. For the *no assist* action, the robot velocity command is zero. To compute the final velocity command a^{total} associated with each move action, the velocity command corresponding to the user's joystick input[2] ${a_h}^{joy}$ is added to the robot velocity command a_r^{vel} with a fixed arbitration factor α of 0.5.

The robot can also explicitly ask the user about his/her goal through question-asking actions ${\mathcal{A}_r}^{qn}$. As asking the user about the low-level target verbally is not feasible, we only include the high-level target in the questions. For a goal g consisting of $\langle o, \theta_o, \mathcal{K}_{\theta_o} \rangle$, the robot action a_g^{qn} will be to ask : "Is OBJECT NAME with GRASP TYPE your goal?". OBJECT NAME and GRASP TYPE will be filled based on o and θ_o. The user can answer either "yes" or "no" through button inputs. Once the user answers "yes", the high-level target is known. At this point, the low-level target is still unknown, as the system still has uncertainty about the user's desired grasp pose for the known high-level target. There are $|\mathcal{G}|$ question-asking actions, one for each goal. In total, there are $2|\mathcal{G}| + 2$ actions. For a better understanding and visualization of the actions selected by the model, please refer to Section IV.

To manage the model's complexity and keep it tractable in scenarios with a large number of goals, we constrain the action space only to consider the top $\mathcal{G}_m$ number of goals according to their probability. If multiple goals have the same probability,

[2]The velocity command corresponding to the user's joystick input is determined using the joy package: http://wiki.ros.org/joy

they are sorted based on their distance to the current gripper pose x, i.e. actions with goals that are closer to the gripper will be considered. This is based on the assumption that the actions associated with the lower probability goals are less likely to yield a high reward, so pruning removes these unpromising actions from the action set. In our experiments, we set $\mathcal{G}_m$ to 10, so the maximum possible number of actions is 22.

The transition function $\mathcal{T}(s, a_r, s')$ models the probability of transitioning from state s to s' after robot action a_r is taken. Since we do not want to reason about low-level velocity commands, we make move actions terminal, similar to [31]. When a question-asking action ${a_r}^{qn}$ is performed, the robot stops moving and the state remains the same. Thus, we need not model the transition function for all actions.

C. Observation Space $\mathcal{Z}$

$\mathcal{Z}$ is the set of observations that the robot can observe at every time step. We represent observation $z \in \mathcal{Z}$ as a tuple $\langle x, {a_h}^{joy}, {a_h}^{qn} \rangle$, where $x \in \mathbb{R}^6$ is the 6-DoF pose of the robot gripper, ${a_h}^{joy}$ is the velocity command corresponding to the joystick input from the human and ${a_h}^{qn}$ is the "yes" or "no" response for question-asking actions.

D. Observation Function

$\mathcal{O}(z, s, a_r) = P(z|s, a_r)$ is the probability of observing $z \in \mathcal{Z}$ after taking action a_r to reach state s. For move actions, the observation is the velocity command corresponding to the user joystick input ${a_h}^{joy}$. Similar to [6], we compute the probability $P({a_h}^{joy}|s = \langle x, g \rangle, {a_r}^{move})$ using principles of maximum entropy inverse optimal control (MaxEnt IOC) [36].

For question-asking actions, the observation is ${a_h}^{qn}$. We assume that the human is truthful and define the probability $P({a_h}^{qn}|s = \langle x, g \rangle, {a_r}^{qn})$ as 0.999 for "yes" answers and 0.001 for "no" answers, if $g_{qn} = g$, where g_{qn} is the goal asked by the question and g is the goal in state s. Similarly, if $g_{qn} \neq g$, the probability is 0.999 for "no" answers and 0.001 for "yes" answers.

E. Reward Function

$\mathcal{R}(s, a_r)$ is the reward function specifying the immediate reward obtained after the robot action a_r is taken in state s. If the action is a move action ${a_r}^{move}$, for a state s with a desired grasp pose k_c and gripper pose x, we want to reward actions that take the gripper closer to the desired grasp pose ($\mathcal{R}_d(s, {a_r}^{move})$) and penalize actions that oppose the user action ($\mathcal{R}_u(s, {a_r}^{move})$). Thus, we define $\mathcal{R}(s, {a_r}^{move}) = \mathcal{R}_d(s, {a_r}^{move}) + \mathcal{R}_u(s, {a_r}^{move})$, where

$$\mathcal{R}_d(s, a_r^{move}) = \begin{cases} \mathcal{R}_t & \text{if } d_{trans} > \delta \\ \mathcal{R}_t + \mathcal{R}_r & \text{if } d_{trans} \leq \delta \end{cases} \quad (1)$$

$\mathcal{R}_t = -100(1 - 2^{-d_{trans} * \beta})$ is the reward based on the Euclidean distance[3] d_{trans}, where β controls the rate of change of reward and is set at 8 in our setup. An exponential function is chosen so that $\mathcal{R}_t$ is bounded at -100. When the gripper is close to the desired grasp pose, or within a δ threshold, an additional reward $\mathcal{R}_r = -100 * \frac{d_{rot}}{2\pi}$ is given. d_{rot} is calculated based on the quaternion distance[3]. We set $\delta = 0.1m$, which is about twice the length of the gripper fingers. This additional rotational reward is only given when the gripper is close to the object to reduce the robot's sensitivity to rotational distances when far, allowing the robot to first approach the correct object. This is similar to our observations on how users teleoperate the arm, which is to translate the arm towards the object first, then adjust the gripper rotation.

$\mathcal{R}_u(s, {a_r}^{move})$ is -5 if the robot velocity command associated with the move action a_r^{vel} opposes the user action ${a_h}^{joy}$. There is opposition if the velocity commands of the robot action and user action are in opposite directions for any dimension i in $\mathbb{R}^6$, i.e. ${}^i{a_h}^{joy} \times {}^i a_r^{vel} < 0$ and $(|{}^i{a_h}^{joy}|, |{}^i a_r^{vel}|) > (0.01, 0.01)$.

To compute $\mathcal{R}_d(s, a_r^{move})$, we need to know the user action $a_{h,t}$ at the next time step t but this is unknown at the time of search. For simplicity, we assume that $a_{h,t-1} \approx a_{h,t}$ since each time step is small, i.e. ≤ 0.3 seconds.

For question-asking actions ${a_r}^{qn}$, a fixed reward of -5 is incurred to encourage the robot to ask a minimal number of questions.

The maximum reward of 100 is given for successfully reaching the user's desired grasp pose. To prevent asking repeated questions and improve the efficiency of the search tree, the maximum penalty of -100 is incurred when the robot asks about a goal that has already been asked about.

F. Belief Tracking

Partial observability of the user's goal is captured using a belief $b(s)$, which is a probability distribution over s. The POMDP maps the belief $b \in \mathcal{B}$ to actions $a_r \in \mathcal{A}_r$, and the solution is an optimal policy $\pi_r : \mathcal{B} \rightarrow \mathcal{A}_r$ that maximizes the value $V_{\pi_r}(b)$ for all $b \in \mathcal{B}$. The value of a policy π_r at belief b is defined as:

$$V_{\pi_r}(b) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, \pi_r(b_t), \pi_h) | b_0 = b\right]$$

where $0 \leq \gamma \leq 1$ is the discount factor set as 0.95 and b_0 is the initial belief. Since the user goal g is the only hidden part in the state s, the probability distribution over s becomes a probability distribution over user goals. Initially, we set a uniform distribution on the possible user's goals. As the user's inputs are observed, the probability distribution on goals is updated using the observation probability function described in Section III-D.

G. Redistributing Grasp Poses

At the start, each goal has a unique high-level and low-level target. The low-level target contains all the grasp poses associated with the high-level target. We assume each goal is equally likely i.e. $p(g) = 1/|\mathcal{G}| \ \forall g \in \mathcal{G}$. As the system

[3] d_{trans} and d_{rot} are calculated between the desired grasp pose k_c and the gripper pose x' reached after applying final velocity command a^{total} corresponding to robot action a_r^{move} to the current gripper pose x.

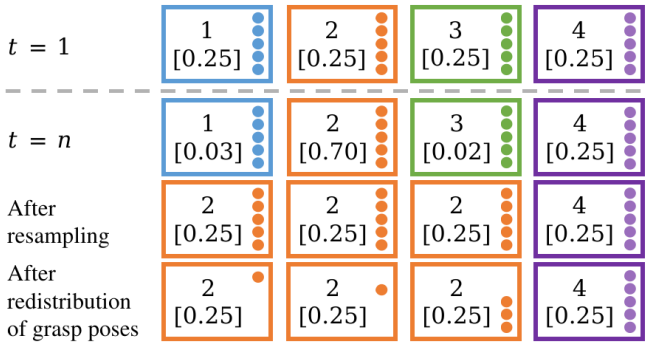


Fig. 2. Redistribution of grasp poses. Each box represents a goal with a high-level target represented by an ID, shown as the upper number in the box, while the lower number denotes the probability of the goal. The dots on the right of the box represent the low-level targets, with each dot representing a grasp pose. For simplicity, we only show five grasp poses in a low-level target in the figure. At $t = 1$, all high-level targets have the same probability (row 1). At $t = n$, the probability of high-level target 2 is high (row 2), making the effective size of belief small. This triggers resampling, which causes high-level target 2 to be duplicated (row 3). After redistributing the grasp poses across low-level targets (row 4), high-level target 2 has three different low-level targets associated with it. The final row shows the state at the end of $t = n$.

observes the user actions, the probability of some goals increases and some goals decreases.

As a standard practice, resampling from the probability distribution is performed when a small number of values (goals in our case) have a much higher probability than other values [37]. The purpose of resampling is to obtain a better estimate of the probability distribution. Resampling is triggered when the effective size $\hat{N}_{\text{eff}}$ of the belief (probability distribution over the goals) measured using Kish's formula [38] is small i.e.

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{|\mathcal{G}|}(p(g_i))^2} < \frac{|\mathcal{G}|}{2}$$

After resampling, a highly likely goal will be sampled multiple times, while a less likely goal might not be sampled. Thus, there will be duplicates of highly likely high-level targets in our set of goals. To consider various grasp poses for these high-level targets, we redistribute the grasp poses in the low-level targets across the duplicated high-level targets. The redistribution of grasp poses in the low-level targets is based on their distance from the current gripper pose x, such that the user's desired pose k_c for each goal is unique. Thus, if a high-level target is duplicated three times, the grasp pose with the smallest distance to the gripper pose will be set as a low-level target for one instance; the grasp pose with the second smallest distance will be set as a low-level target for the second instance; the list of remaining grasp poses will be set as the low-level target for the third instance, as illustrated in Fig. 2.

Thus, after redistributing the grasp poses, there will be more goal-oriented move actions towards the highly likely high-level targets, with each action considering a different grasp pose. This allows the POMDP solver to reason which grasp pose for the same object and grasp type is the user's target pose, which is important especially after a question is asked.

IV. Simulation Experiments

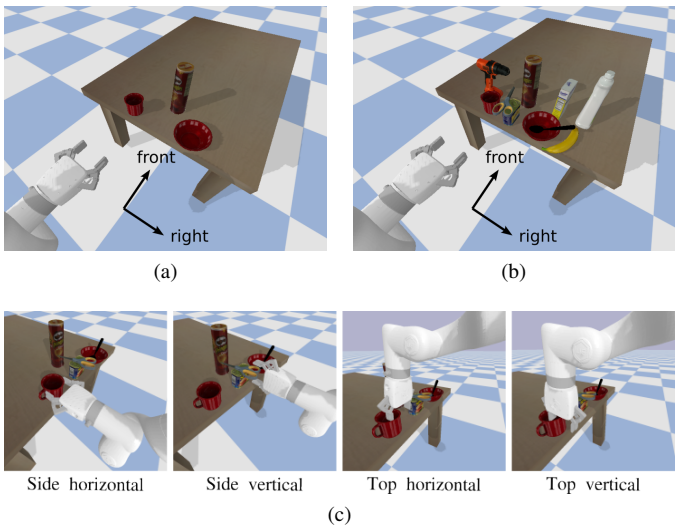


Fig. 3. (a) Scene 1 with 3 objects and 4 goals in simulation. (b) Scene 2 with 10 objects and 18 goals in simulation. (c) Grasp types used in Scene 2.

To evaluate different components of our method (referred to as *Policy*), we first conduct experiments in simulation to ensure the object poses and grasp poses are the same when comparing different components. hyp-DESPOT [22] was used to solve the POMDP for *Policy* with a planning time of 0.1 seconds. All tasks in this section were performed by one of the authors.

For comparison, we use a state-of-the-art shared autonomy approach based on passive information gathering [6] (referred to as *HO*) as our baseline policy. To better study the importance of the *no assist* action and *question-asking* actions, we also implemented a no assist version of the baseline *HO* (referred to as *HO+NoAssist*). In *HO+NoAssist*, when the optimal robot action computed by *HO* does not oppose the user action, we follow the implementation in the original paper, i.e. the robot action is executed. However, when the robot action opposes the user action, we provide the user full control, i.e. we only execute the user action. Opposition is measured in the same way as described previously in Section III-E. For both *HO* and *HO+NoAssist*, the goals are set to be the same as the high-level targets in our method.

First, we discuss the importance of different actions in the POMDP in Section IV-A by studying in detail how the system behaves for a given task when different actions are not included. Next, we demonstrate the robustness of our method by comparing it with *HO* and *HO+NoAssist* for 8 different tasks in simulation IV-B.

The following objective measures were used to evaluate the performance of the algorithm:

- **Task completion time** measures the time taken to grasp an object.
- **Number of mode changes** records the number of times the mode was switched between translation and rotation, measuring the effort required to achieve the goal.

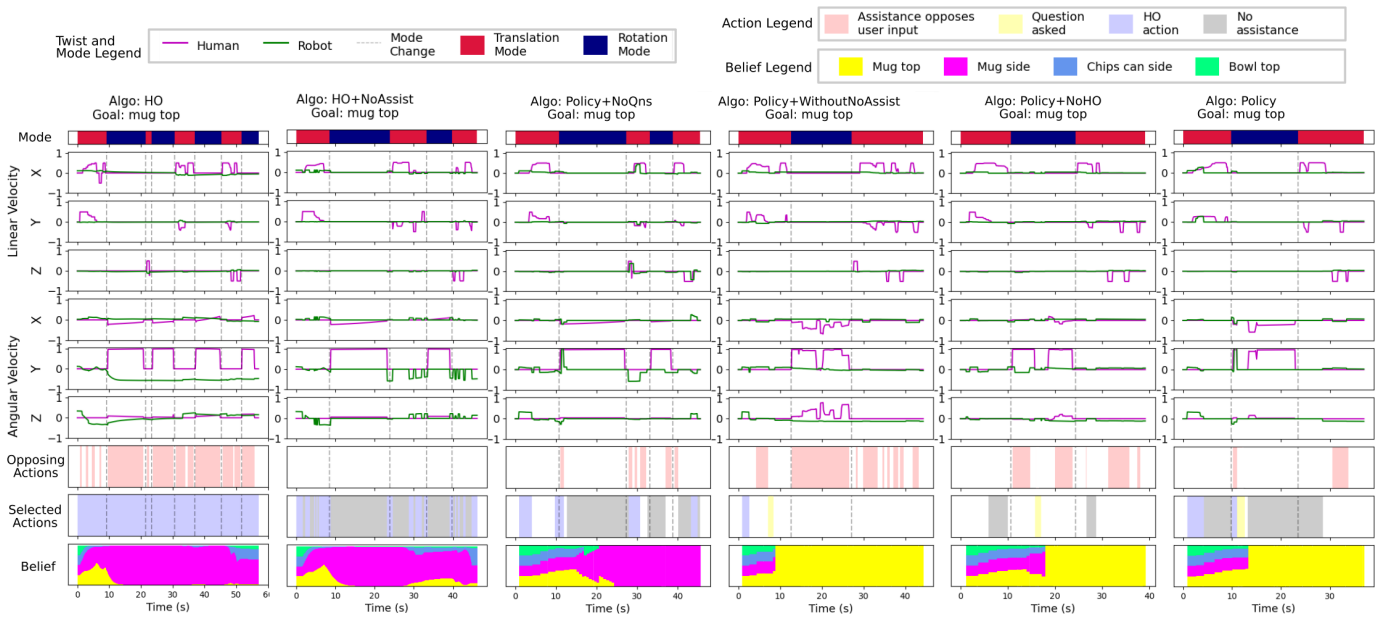


Fig. 4. Trajectories, actions selected and belief for *HO*, *HO+NoAssist*, *Policy+NoQns*, *Policy+WithoutNoAssist*, *Policy+NoHO* and *Policy* for a sample run to grasp a cup with a top grasp in simulation.

- **Total joystick input** measures the total joystick input provided by the user, computed by the absolute sum of the raw joystick input in 3 dimensions at every 0.025s.
- **Percentage opposing time** measures the percentage of time where the robot actions oppose the user actions, which measures the assistance provided. Opposing actions are defined according to Section III-E.
- **Correct prediction rates** records the proportion of trials where the system predicted the user's goal correctly at the end of the task with a confidence greater than 0.7.

A. Importance of Different Actions in Model

To demonstrate the importance of different high-level actions, we use simulation Scene 1 (Fig. 3a) and perform the same task of grasping a cup from the top with different ablated versions of our method and the two baselines. The different ablated versions of our algorithm are as follows:

- *Policy+NoQns*: Our method without *question-asking* actions
- *Policy+WithoutNoAssist*: Our method without *no assist* action
- *Policy+NoHO*: Our method without *hindsight optimization* action

For each method, we plot the belief change, the actions generated, opposing actions, the angular and linear velocity commands of the robot and user, and the mode changes in Fig. 4. The tasks in this section were only performed once for a detailed analysis of that particular instance. The results for the objective measures are shown in Table I.

The following sections discuss the importance of different high-level actions based on the quantities plotted in Fig. 4 and measured in Table I.

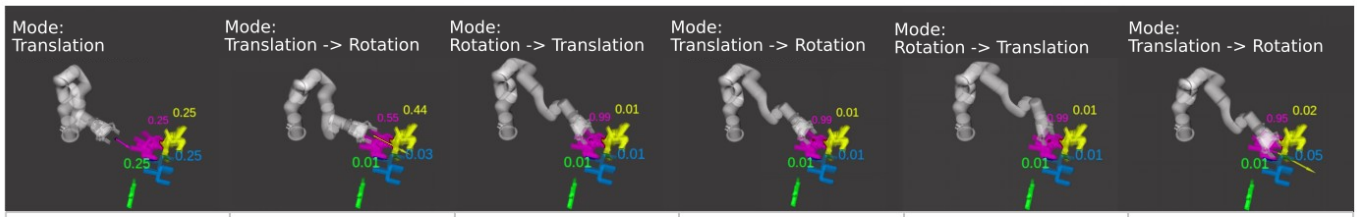
1) No Assist Action: We evaluate the importance of the *no assist* action by comparing *Policy* against *Policy+WithoutNoAssist*. The number of opposing actions is significantly higher for *Policy+WithoutNoAssist* (Fig 4). This also led to an increase in the opposing time, total joystick input and task completion time (Table I).

A similar effect is seen when comparing *HO* against *HO+NoAssist*. Note that there are no opposing actions in *HO+NoAssist* by design, as the robot action is not executed whenever it opposes the user action. Thus, the assistance provided in *HO+NoAssist* is less than *HO*, which should have led to more joystick inputs from the user. Instead, *HO+NoAssist* performed better in terms of total joystick input, task completion time and the number of mode changes, as seen in Table I. This is because the *no assist* action prevents the system from providing the wrong assistance.

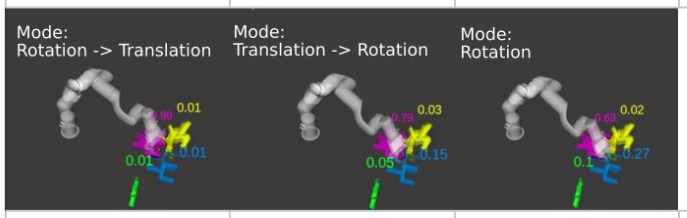
Thus, for both *Policy* and *HO*, including a *no assist* action reduces the opposing actions, which helps improve the overall assistance.

2) Question-Asking Actions: First, we compare the belief change for algorithms with *question-asking* actions[4] (*Policy*, *Policy+NoAssist*, *Policy+NoHO*) and algorithms without (*HO*, *HO+NoAssist* and *Policy+NoQns*). From Fig 4, all three algorithms without *question-asking* actions failed to converge to the correct goal. Even though the task could be accomplished despite the incorrect belief convergence, there were more mode changes and total joystick input compared to the algorithms with *question-asking* actions (Table I).

[4]The timestamp when question-asking is triggered varies across different trials. This is because the POMDP model takes into account the user action and the possible robot actions when determining whether information-gathering is the best action for that time step. Given that user inputs vary across different trials, the possible move-to-goal robot actions also vary due to a difference in the gripper state. Thus, information-gathering actions are selected at different time steps for different trials.

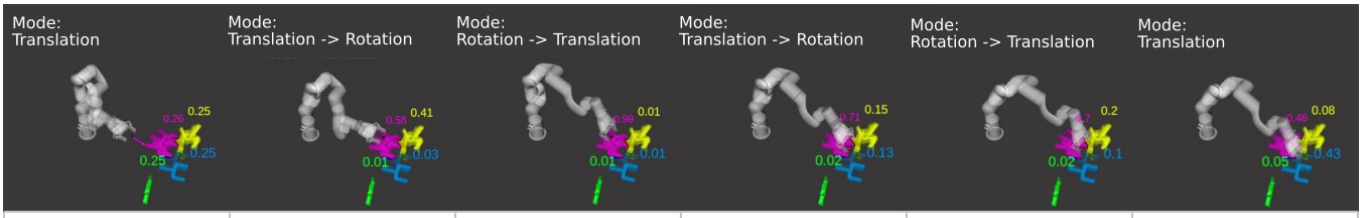


1. Start	2. User translates gripper towards cup	3. User rotates gripper to reach a top grasp	4. User translates gripper to fine-tune position, but robot rotates gripper towards a side grasp	5. User rotates gripper back to a top grasp.	6. When user is translating gripper, robot rotates gripper back to a side grasp again



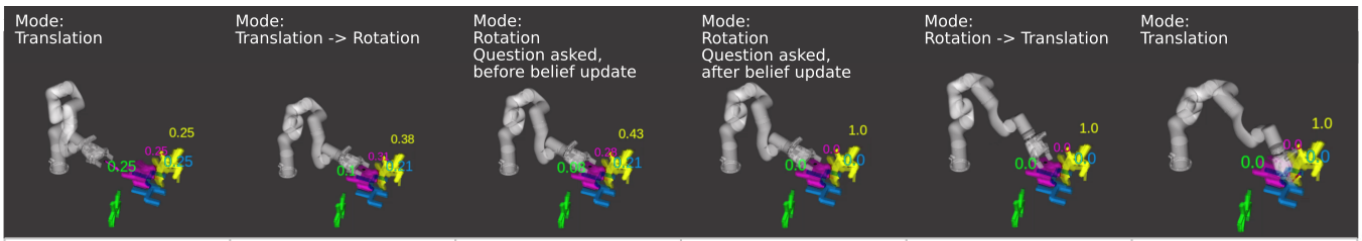
7. User rotates gripper back to a top grasp.	8. User translates gripper to fine-tune position.	9. User closes the gripper to grasp object.			

(a) Algorithm: *HO*



1. Start.	2. User translates gripper towards cup.	3. User rotates gripper to reach a top grasp.	4. User translates gripper to fine-tune position, but robot rotates gripper towards a side grasp.	5. User rotates gripper back to a top grasp.	6. User translates gripper to fine-tune position and closes the gripper to grasp

(b) Algorithm: *HO+NoAssist*



1. Start.	2. User translates gripper towards cup.	3. Under uncertainty, system asks the question "Is mug with top grasps your goal?"	4. User answers "yes", belief is updated. System starts rotating gripper to a top grasp.	5. User further rotates gripper to a top grasp.	6. User translates gripper to fine-tune position and closes the gripper to grasp

(c) Algorithm: *Policy*

Fig. 5. Screenshots showing the belief, gripper pose and teleoperation mode for the runs in simulation to grasp the cup with a top grasp (in yellow) for (a) *HO* (b) *HO+NoAssist* (c) *Policy*. A different colour is used to show each high-level target, and the corresponding belief for each high-level target is displayed using the same colour. For each algorithm, screenshot 1 shows the starting pose while the last screenshot shows the final grasping pose where the gripper is closed. The screenshots in between are taken before a mode change. In *Policy*, screenshot 3 is taken when a question is asked, while screenshot 4 is taken to show the belief update after the user answered "yes".

TABLE I
RESULTS FOR SCENE 1 GOAL 1

Algorithm	Completion Time (s)	Mode Changes	Total Joystick Input	Opposing Time (%)	Correct Prediction (T/F)
Manual	42.73	4	316.45	-	-
HO	57.13	7	471.81	74.6	F
HO+NoAssist	46.27	4	347.96	-	F
Policy+NoQns	45.72	4	331.94	10.7	F
Policy+WithoutNoAssist	44.42	2	286.61	51.7	T
Policy+NoHO	39.23	2	223.62	25.4	T
Policy	**37.06**	2	**218.45**	**9.7**	T

To understand why the belief does not converge to the correct goal with passive information gathering in *HO* and *HO+NoAssist*, we show the screenshots of sequences of actions for *HO*, *HO+NoAssist* and *Policy* in Fig. 5. Each screenshot shows the gripper position and the probability of different goals before a mode change, or before and after a question is asked.

For *HO* (Fig. 5a), as the user translates the gripper towards the cup, both the top and side grasps for the cup have high probabilities in screenshot 2. At this point, the user performs a mode change. When the user rotates the gripper, the gripper is closer to the side grasp, which increases the probability of the cup with side grasp. Thus, the action computed by *HO* opposes the user action. This is also captured in Fig 4, under the Y-axis angular velocity and the opposing actions for *HO*. To overcome the wrong assistance, the user performs several mode changes, but the belief has converged to the wrong goal of a cup with side grasp. As such, the robot always assists towards the cup with side grasp and opposes the user.

For *HO+NoAssist* (Fig. 5b), similarly, the top and side grasps for the cup have high probabilities in screenshot 2. The probability of the cup with side grasp again increases when the user switches to rotation mode as the gripper is closer to the side grasp. However, the robot does not oppose the user action in this case, which can be seen in Fig. 4, where the *no assist* action is selected while in rotation mode for *HO+NoAssist* algorithm. However, when the user switches to translation mode to fine-tune the gripper position, the robot starts rotating the gripper to a side grasp due to a wrong belief. This is because rotating the gripper does not oppose user actions in translation mode. Thus, the user has to switch modes again to rotate the gripper to the correct orientation before grasping the cup.

For *Policy* (Fig. 5c), a question is asked in screenshot 3, when both the top and side grasps of the cup have a high probability. Since the user answers "yes" to the question, the high-level target is known. The robot assists the user towards the correct goal and the task is completed with minimal time, mode changes and total joystick input, as shown in Table I. We note that there is still uncertainty over the user's desired grasp pose (low-level target), so opposing actions still exist after question-asking, as shown in Fig. 4.

For the same task, we also present a scenario where the system reasons to ask a question about the cup with side grasps instead of top grasps (Fig. 6). Even though the user answers "no", the system receives sufficient information to disambiguate between the goals and could provide correct assistance without asking additional questions.

This analysis shows the need for active information gathering, especially when the goals are not well separated, as it is difficult to estimate the user's intent from passive observation. However, it is not trivial to decide *when* and *which* questions to ask. One way could be to ask a question whenever the entropy of the belief is higher than a certain threshold. However, this will lead to questions being asked at the start about a random goal since all goals are equally likely. In some cases, when the policies for all goals are similar, active information gathering might not even be needed. A threshold-based method will still ask questions as it does not evaluate the results of possible future actions. By modelling the problem as a discrete-action POMDP, we can compute the optimal action for different cases in a principled manner.

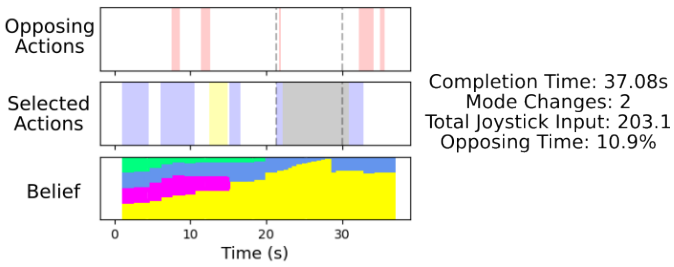


Fig. 6. Actions selected, belief and objective measures for a sample run using *Policy* in simulation to grasp a cup with a top grasp. The legend is the same as in Fig. 4. In this run, the question "Is mug with side grasps your goal?" was asked and the user answered "no". No additional questions were asked after that.

3) Hindsight Optimization Action: To evaluate the importance of including an action that assists towards a distribution of goals, we compare *Policy* against *Policy+NoHO*. From Fig. 4, *Policy* selects the *hindsight optimization* action at the start while *Policy+NoHO* selects actions moving to different goals. Table II shows the detailed sequence of actions selected by *Policy+NoHO* and *Policy* before a question is asked. Different move-to-goal actions are selected by *Policy+NoHO*, e.g. moving to the mug with top grasps at the first time step, but moving to the mug with side grasps at the second time step. While this does not cause any opposition when the user is in translation mode, the robot action would change the gripper's orientation, which could confuse the user. The random selection of goals at the start leads to a slightly longer task completion time and more joystick input (Table I). In our experiments, we observed that the *hindsight optimization* action is often selected at the start, suggesting that this is an

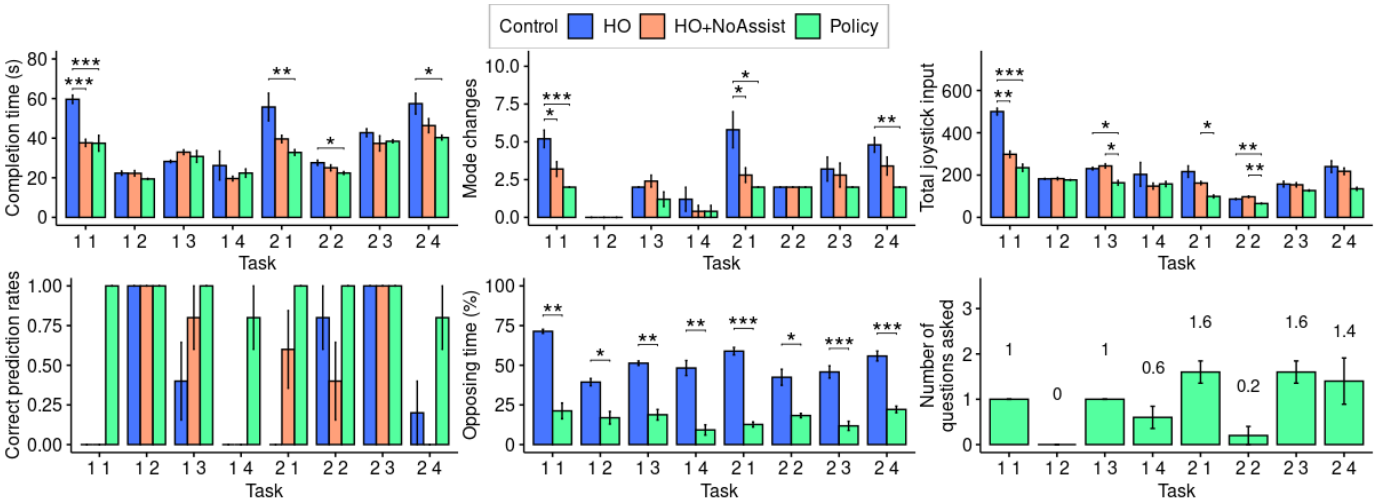


Fig. 7. Results for the qualitative experiments in simulation, where *= $p < 0.05$, **= $p < 0.01$, ***= $p < 0.001$, ****= $p < 0.0001$.

optimal action at the start when the goals are far and there is a minimal policy that is good for all goals. By including this action in the POMDP model, the solver can choose a more optimal action instead of selecting random actions.

TABLE II
ACTIONS SELECTED

Time step	Algorithm	
	Policy+NoHO	Policy
1	Move to mug with top grasp	HO action
2	Move to mug with side grasp	HO action
3	No assist	No assist
4	No assist	No assist
5	Move to bowl with top grasp	No assist
6	Move to mug with side grasp	HO action
7	Move to mug with side grasp	Ask about mug with top grasp
8	Move to mug with side grasp	
9	Move to mug with top grasp	
10	Ask about mug with top grasp	

TABLE III
HIGH-LEVEL TARGETS FOR EACH TASK

Scene	Task	High-Level Target
1	1	Cup with top grasps
1	2	Chips can with side grasps
1	3	Bowl with top grasps
1	4	Cup with side grasps
2	1	Cup with top horizontal grasps
2	2	Scissors with side vertical grasps
2	3	Drill with side vertical grasps
2	4	Spoon with top vertical grasps

B. Robustness

After discussing the importance of various components of our approach through in-depth analysis of a single task, we demonstrate the robustness of our method by evaluating its performance for 8 different tasks (Table. III) across 2 different simulation scenes (Fig. 3). Scene 1 only contains 3 objects, a cup, a chips can and a bowl, giving 4 high-level targets – cup with top grasps, cup with side grasps, chips can with side grasps, and bowl with top grasps. We also set up a more complex scene, Scene 2, which consists of 10 objects and 18 high-level targets. We define four grasp types to generate more goals in Scene 2, as shown in Fig. 3c. The high-level targets in Scene 2 are cup with top horizontal grasps, cup with side horizontal grasps, chips can with side horizontal grasps, meat can with side horizontal grasps, scissors with top vertical grasps, scissors with side vertical grasps, drill with top vertical grasps, drill with side vertical grasps, drill with side horizontal grasps, bleach cleanser with side horizontal grasps, spoon with top vertical grasps, spoon with side vertical grasps, spoon with side horizontal grasps, bowl with top vertical grasps, bowl with top horizontal grasps, banana with top horizontal grasps, sugar box with top horizontal grasps and sugar box with side horizontal grasps.

We compare our method *Policy* with the baseline *HO* and *HO+NoAssist* for these 8 tasks. For evaluation, one of the authors performed each task 5 times for each algorithm and the objective measures were recorded. In addition, we also note the average number of questions asked for each task for *Policy*. The results of this evaluation are presented in Fig. 7.

The difficulty of tasks is reflected in the task completion time and the number of mode changes for the baseline *HO*. Task 2 in Scene 1 was the easiest task, requiring no mode changes and was completed in the shortest time. All three methods were able to predict the intent correctly and performed similarly. Task 1 in Scenes 1 and 2, and Task 4 in Scene 2 required a longer task completion time and a greater user effort. For these tasks, *HO+NoAssist* performed better than *HO* in terms of the number of mode changes and total joystick input, while *Policy* outperformed *HO+NoAssist*. This shows that our approach performs as well as *HO* and *HO+NoAssist* for simple tasks and provides better assistance when the task is difficult.

While the system asked a relatively higher number of questions in Scene 2, the highest number of questions asked only averaged to be 1.6 in Tasks 1 and 3. This shows that *Policy* can reason *whether* to ask questions and can scale to more complex scenes.

The simulation experiments allowed us to perform an in-

depth analysis of various components and study the robustness of our method by performing different tasks in a controlled environment. To study whether our approach has a practical impact on usability and efficiency for real users, we conducted a user study using a real robot arm with human subjects.

V. User Study on Real Arm – Simple Scene

We conducted a within-subject study on three reaching and grasping tasks. A total of 18 participants (15 without motor impairments, 2 stroke subjects, 1 Duchenne muscular dystrophy (DMD) subject; 7 female 11 male; age range 15 - 60) were recruited. All participants gave their signed consent for the experiment procedure, which was approved by the Nanyang Technological University Institutional Review Board.

To ensure the experiment was not too long and to prevent user confusion when assessing different algorithms, we compare our algorithm *Policy* with the existing approach *HO*. Users also tried a *manual* approach before trying *Policy* and *HO* so that they could evaluate whether the robot assistance helped them complete the tasks. We did not compare against a fully autonomous approach as past studies have shown that shared autonomy systems outperform these approaches and that users prefer to retain control [6], [14], [39].

A. Hypotheses

We tested the following hypotheses in our experiments:

H1 Active information gathering will reduce the time taken to complete the task during teleoperation.

H2 Active information gathering will reduce the effort required to teleoperate the robot arm.

H3 Active information gathering in the form of asking questions will improve intent recognition and reduce opposing actions from the robot.

H4 Participants will prefer *Policy* over *HO*.

B. Experimental Setup

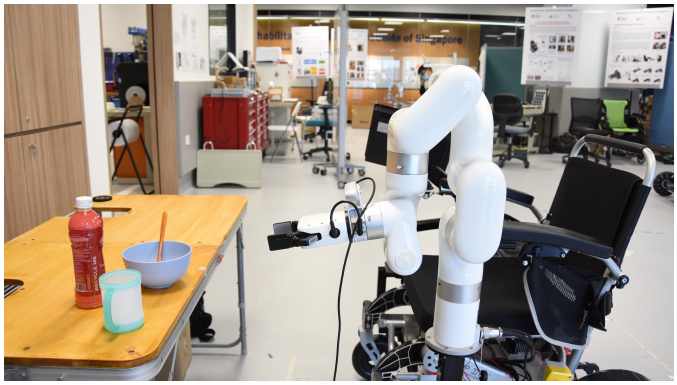
The experiments were set up with four objects on the table – a cup, bottle, bowl and spoon. The objects were placed in the same positions to ensure consistency across subjects. The objects were placed close to each other without occlusion from the camera (Fig. 8a), an Intel Realsense Depth Camera D435 mounted on the xArm-6 gripper.

Mask R-CNN [34] was used to detect the objects in the scene. We assumed the set of possible object classes the user might want to grasp is known. Based on this assumption, we removed objects like tables and chairs detected.

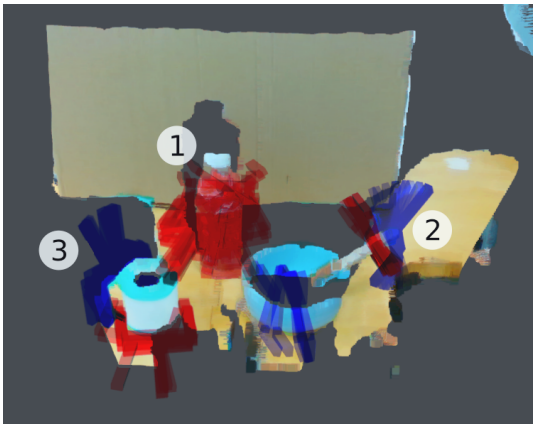
For both shared autonomy algorithms, the Grasp Pose Detection (GPD) package [35] was used to generate grasp poses, which were classified into different goals – top grasps and side grasps[5]. In total, there were 6 goals generated (Fig. 8b).

The xArm-6 robotic arm (UFactory) with the xArm two-finger gripper was used. The robotic arm was mounted on a wheelchair that was fixed during the experiment. Users teleoperated the robotic arm using a Razer Hydra controller, with the right joystick and the left and right trigger buttons controlling three degrees of freedom (DoFs). Three buttons on the right controller were used to switch control modes, control the gripper closing/opening and answer "yes" or "no" for question-asking actions. To allow the stroke subjects to control the arm using only their healthy hand, the joystick controls were modified such that the stroke subjects had a 3-DoF joystick on the right controller. For all subjects, there were 2 control modes available – translation and rotation mode. The rotational component of the velocity command was set to zero in translation mode and vice versa.

hyp-DESPOT [22] was used to solve the POMDP for *Policy*. The POMDP model had 14 actions and a planning time of 0.3 seconds. The robot communicated through a speaker, using an off-the-shelf text-to-speech package to ask questions.



(a)



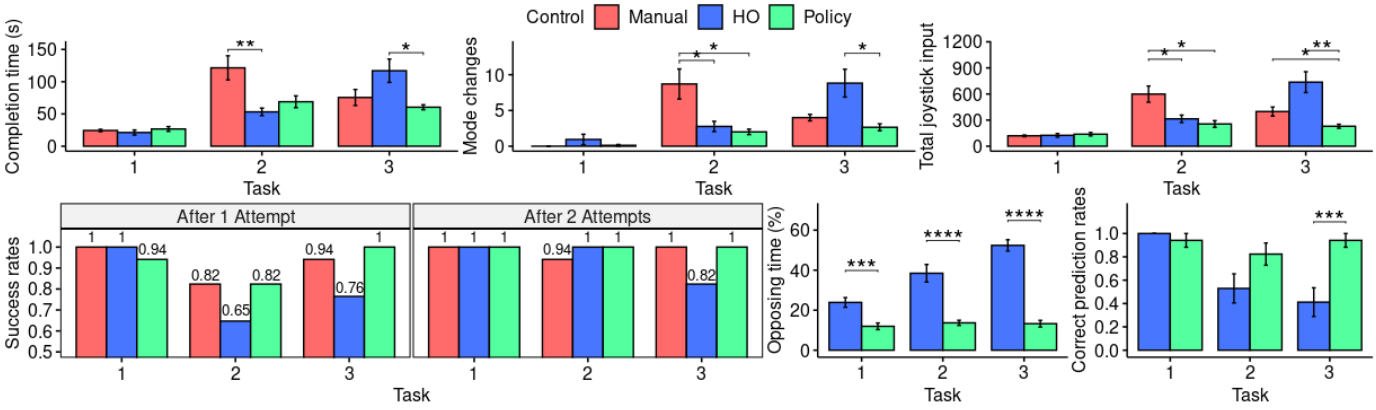
(b)

Fig. 8. (a) Experimental setup with four objects placed on the table. The robot arm was reset to this pose before each trial. (b) Goals generated in our experiments. Goals for each object are shown in a separate colour, where top grasps are shown in blue and side grasps are shown in red. There are 6 goals[5] in total. Participants had to complete three tasks by grasping the object with a particular grasp type in the following order - 1) bottle with side grasp, 2) spoon with top grasp and 3) cup with top grasp, as indicated by the numbers in the image.

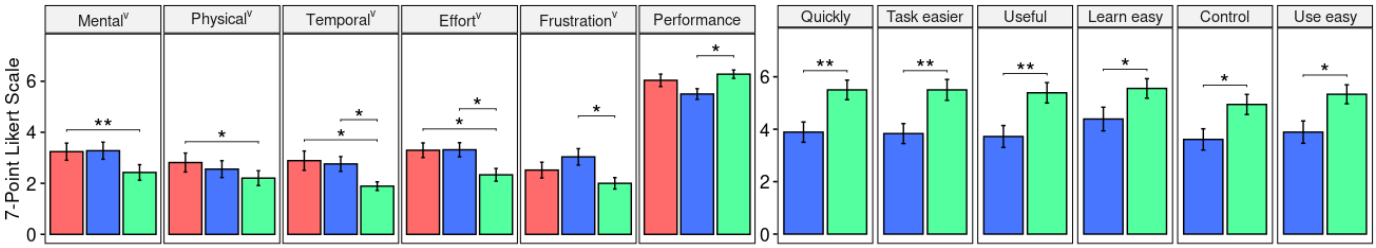
C. Procedure

Participants had to complete three trials, one each for *manual, HO* and *Policy*. Each trial consisted of the following

[5]Note that the bottle only had side grasps as top grasps exceeded the arm's workspace, while the bowl only had top grasps since the gripper could not grasp the bowl with a side grasp.



(a)



(b)

Fig. 9. Bar plots with error bars for each method across all participants for the simple scene user study. The plots for (a) objective measures and (b) subjective measures, where *= $p < 0.05$, **= $p < 0.01$, ***= $p < 0.001$, ****= $p < 0.0001$. $^\vee$ denotes that a lower score is better. The colour of the bars representing the control methods is consistent across all graphs.

three tasks (Fig. 8b):

1) Grasp the bottle with a side grasp;
2) Grasp the spoon with a top grasp;
3) Grasp the cup with a top grasp.

The tasks were chosen to include tasks of different difficulties. The first task of grasping the bottle in each trial was the easiest since it did not require rotation of the gripper. The second and third tasks were more difficult as users had to rotate the gripper towards a top grasp and the system had to disambiguate between top and side grasps for the object. For each task, participants had to teleoperate the robotic arm from a fixed starting pose and grasp the target object with the specified grasp type. The gripper closing signalled the end of the task for task completion time measurement, but subjects were asked to lift the objects after grasping. If users failed to grasp or lift an object, they were given an additional attempt. The robotic arm and scene were reset after each task for consistency.

Before the trials, participants were given a user sheet containing instructions, task descriptions and controller inputs. They were given five minutes to practice manual teleoperation on a practice scene to familiarise themselves with the controls. The scene in Fig. 8a was then set up and participants completed all the tasks with *manual* control. *Manual* control was always performed first so participants could evaluate the robot assistance using the shared autonomy algorithms later. Similarly, for the shared autonomy methods, participants had five minutes to practice on a test scenario. They then completed all the tasks with the same control strategy before switching to the other shared autonomy method. The order of testing the shared autonomy methods was randomized across participants to counteract the effects of novelty and practice.

After each task, participants filled out a **NASA-TLX** [40] survey, where participants had to rank their workload on six sub-scales. **Mental** demand measured the mental and perceptual activity required; **physical** demand measured how physically demanding or laborious the task was; **temporal** demand measured how much time pressure the participant felt; **effort** measured how hard the participant had to work to achieve the level of performance; **frustration** measured how annoyed, stressed and irritated the participant felt; **performance** measured how successful the participant felt in completing the task.

For the two shared autonomy methods, they were also asked to rank their agreement on a 7-point Likert scale after each trial for the following **usefulness** and **ease-of-use** statements.

1) "This algorithm helped me complete the task more **quickly**."
2) "This algorithm made it **easier** to complete the **task**".
3) "If were to teleoperate a robotic arm, this algorithm would be **useful** for me."
4) "**Learning** to use this algorithm is **easy** for me."
5) "I felt in **control** while using this algorithm."
6) "I would find this algorithm **easy** to **use**."

At the end of all three trials, participants were also asked which method provided them with **better assistance** and which method they **preferred**.

We also evaluated the performance of the three control

methods using objective measures mentioned in Section IV and included **success rates** as an additional measure, which is the proportion of successful trials. The trial is successful if the participant grasps the object with the intended grasp type, lifts the object, and none of the objects falls off the table.

D. Results

Since the subjects with upper limb motor impairments did not have difficulties operating the joystick, we combined the analysis for all the subjects together.

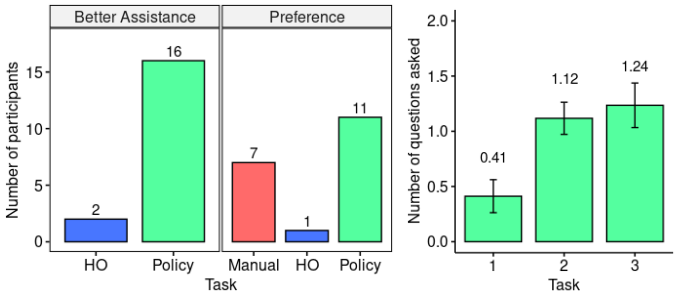
The experimental results reflected the varying difficulty levels of the tasks. Task 1 did not require rotation of the gripper pose, resulting in fewer mode changes and faster task completion time. Note that for **task completion time**, **mode changes**, **total joystick input** and **opposing time**, the data reported is from the first attempt if the attempt was successful. Otherwise, the data reported is from the second attempt if the second attempt was successful. If both attempts were unsuccessful, the data was not included in the analysis of these measures.

1) Objective Measures: A two-way repeated measures ANOVA was performed to evaluate the effect of the task and assistance method (*manual, HO, Policy*) on task completion time, number of mode changes and success rates. A Greenhouse-Geisser correction was performed if the data violated the assumption of sphericity. If a significant main effect was found, a post hoc analysis using pairwise comparisons was performed using Bonferroni corrections to identify which conditions were statistically different from each other (Fig. 9a).

For **task completion time**, there was a statistically significant difference between control methods ($F(2, 31) = 7.13, p < 0.001$). Post hoc analysis showed a significant difference between *HO* and *manual* ($p = 0.006$) in Task 2 and between *Policy* and *HO* ($p = 0.019$) in Task 3. There was a marginal difference between *Policy* and *manual* ($p = 0.093$) in Task 2. Even though insignificant, we note that *Policy* had a slower task completion time compared to *HO* in Task 2. This additional time can be attributed to the time to ask and answer questions. Thus, *Policy* performs at least as well as or better than *HO* and *manual* in terms of task completion time, showing support for **H1**.

Number of mode changes differed significantly between control methods ($F(2, 36) = 8.039, p < 0.001$). Post-hoc analysis revealed significant differences between *Policy* and *manual* ($p = 0.014$) and between *HO* and *manual* ($p = 0.035$) for Task 2, and between *Policy* and *HO* ($p = 0.014$) for Task 3. Marginal differences were also found between *HO* and *manual* ($p = 0.099$) for Task 2.

Total joystick input differed significantly between control methods ($F(2, 34) = 10.625, p < 0.001$). Post-hoc analysis revealed significant differences between *Policy* and *manual* ($p = 0.012$), and between *HO* and *manual* ($p = 0.025$) for Task 2, and between *Policy* and *HO* ($p = 0.003$) and between *Policy* and *manual* ($p = 0.041$) for Task 3. Marginal differences were also found between *HO* and *Policy* ($p = 0.071$) for Task 2. Thus, the **number of mode changes** and **total joystick input** measures show support for **H2**.



(a) (b)

Fig. 10. (a) User preferences when asked which method provided better assistance and which method they preferred the most. (b) Number of questions asked by *Policy* for each task for the simple scene user study.

For **success rates**, there were multiple failures due to grasping too early, knocking over objects and opposition from the assistance. There were 18 failures on the first attempt – for Task 1, there was 1 failure with *Policy*; for Task 2, there were 3 failures with *manual*, 6 failures with *HO* and 3 failures with *Policy*; for Task 3, there was 1 failure with *manual* and 4 failures with *Policy*. Participants who failed the first attempt were given a second attempt. After which, there was 1 failure with *manual* for Task 2 as the bowl fell off the table after the spoon was grasped, and 3 failures with *HO* for Task 3 because of opposition from the control method due to incorrect intent prediction.

A Wilcoxon signed-rank test was performed on the assistance measures, comparing the **percentage opposing time** provided by *Policy* and *HO*. A statistically significant difference was found for all tasks, suggesting that *Policy* produces fewer robot actions that oppose user actions. Even for a simple task like Task 1, where users did not need to change the gripper orientation, *Policy* had a significantly lower percentage opposing time ($p < 0.01$). For tasks that required changing the gripper orientation, users spent a significantly larger proportion of time opposing the robot action in *HO* ($p < 0.001$). Since the robot should minimize conflicts with the user in shared autonomy [4] for effective assistance, **H3** is supported.

For **correct prediction rates**, *HO* predicted all trials in Task 1 correctly, but *Policy* performed better in Tasks 2 and 3, with a statistically significant difference for Task 3. This suggests that for a simple task like Task 1, both *HO* and *Policy* can predict the user intent accurately, but *HO* struggles in more complex tasks. This further supports **H3**.

2) Subjective Measures: For self-reported participant workload using the **NASA-TLX** survey, a Friedman's test was performed. Significant differences were found for all measures (Fig. 9b), with *Policy* outperforming *HO* and *manual*. A post hoc analysis was performed using Bonferroni corrections.

A Wilcoxon signed-rank test was performed on the user-reported subjective measures on the **usefulness** and **ease-of-use** of the shared autonomy algorithms. Significant differences were found for all subjective measures between *Policy* and *HO* (Fig. 9b), showing that users felt that *Policy* helped them achieve the task more quickly, performing the task was easier, the assistance was useful and they felt in control.

When asked to select the algorithm that provided **better assistance**, 88.9% of participants selected *Policy*, supporting **H4** (Fig. 10a). Most participants preferred *Policy* over *HO* as the "questions asked allow communication between operator and robot" and that they could "better understand what the algorithm is trying to do". With better intent recognition, many participants also commented that they did not have to "fight against the robot". Notably, all three subjects with upper limb impairment preferred *Policy*, commenting that for *HO*, "the robot doesn't know what I want" and the robot "didn't move the way I expected it".

When asked about their **preference** between all three control methods, 9 participants selected *Policy*, 5 participants selected *manual* and only 1 participant selected *HO*. There were 2 participants who selected both *Policy* and *manual*, and 1 participant who did not have any preference. Participants that preferred *manual* commented that *manual* provided them with the greatest control, suggesting there is further room for improvement for *Policy*.

E. Discussion

Generally, the results show that *Policy* can assist users towards their goals at least as well, or better than *HO* and *manual*, validating our hypotheses. For a simple task that does not require any gripper rotation like Task 1, all three control methods perform similarly.

Our experimental results show the benefits of active information-gathering actions in the form of asking questions, which improves intent recognition. The average number of questions asked for each task was 0.41, 1.12 and 1.24 respectively (Fig. 10b), showing that the model does not ask too many questions and can reason if and when to ask questions.

In particular, for Task 3, where *HO* cannot predict the user's goal accurately, information-gathering actions in *Policy* resulted in fewer mode switches and shorter task completion times. Given that the same planner is used to generate robot actions for both methods, the difference in performance can be attributed to the information-gathering actions. Interestingly, with incorrect goal inference, we found that *HO* performed worse than *manual* for Task 3 in terms of task completion time and number of mode changes. This supports that no assistance is better than providing wrong assistance.

VI. User Study on Real Arm – Complex Scene

To investigate the scalability of the proposed approach to scenes with a larger number of goals, we conducted another within-subject study on three reaching and grasping tasks on a more complex scene. 15 participants (6 female 9 male; age range 21 - 60) were recruited from the Nanyang Technological University community. This study was approved by the Nanyang Technological University Institutional Review Board.

A. Hypotheses

In addition to the hypotheses in Section V-A, we tested the following hypothesis:

H5 The average number of questions asked in the complex scene will not increase significantly as compared to the simple scene.

(a)

Task 1 - Bowl with top horizontal grasps

Task 2 - Can with side horizontal grasps

Task 3 - Drill with side vertical grasps

(b)

Fig. 11. (a) Experimental setup with 11 objects placed on the table. (b) The three tasks in this user study – grasping the object with a particular grasp type in the stated order.

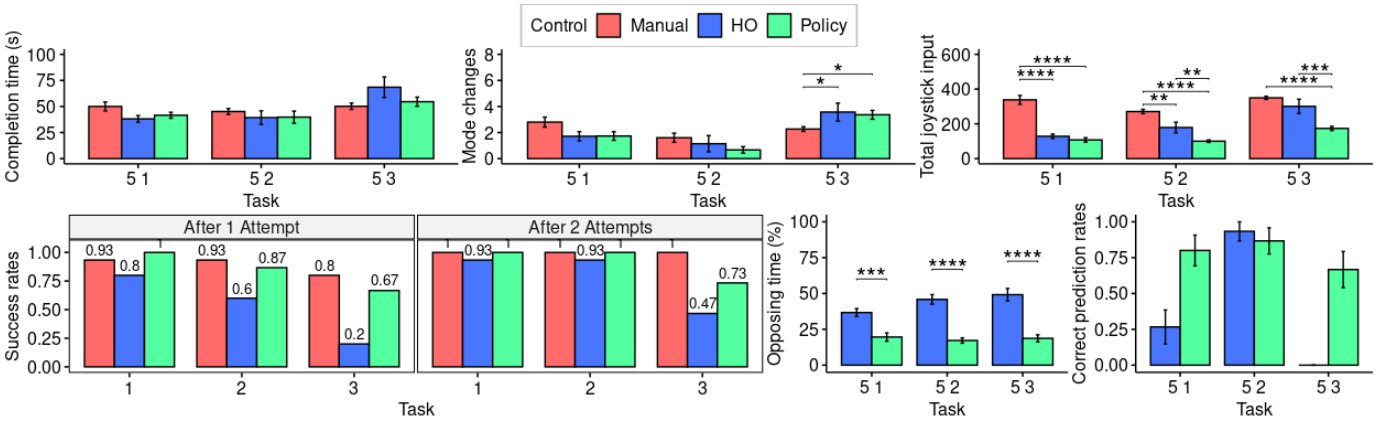
TABLE IV
Goals in Complex Scene User Study

Object	Grasp Types
Drill	Top vertical, side horizontal, side vertical
Apple	Top vertical, top horizontal
Banana	Top horizontal
Plate	Top vertical
Scissors	Top horizontal, side vertical
Spray cleaner	Top vertical, side vertical, side horizontal
Cup	Top vertical, top horizontal, side horizontal
Bottle	Side horizontal
Can	Top vertical, side horizontal
Bowl	Top vertical, top horizontal
Spoon	Top vertical, side vertical

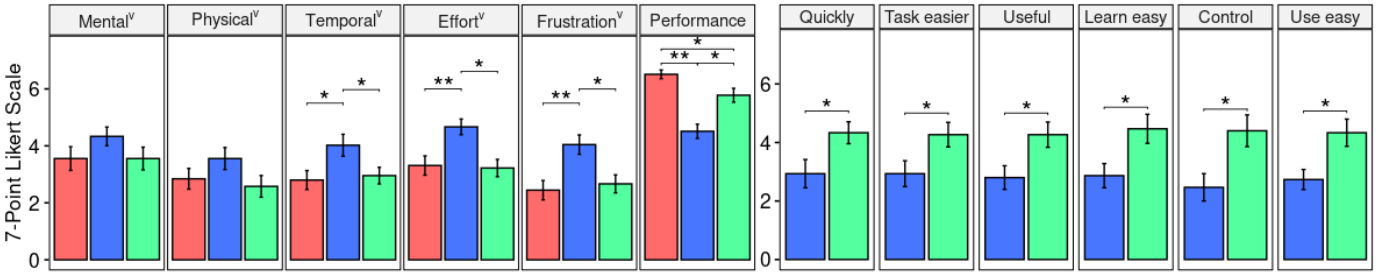
B. Experimental Setup

The same hardware, i.e. robotic arm, camera and joystick controller, was used as in Section V. 11 objects were placed on the table in the scene (Fig. 11a, Table IV). To generate more goals, we defined four grasp types – side horizontal, side vertical, top horizontal and top vertical grasps, as shown in Fig. 3c.

As some of the objects could not be detected using Mask R-CNN [34] with the MS COCO dataset [41], we manually labelled the object names in the scene. Using the Segment Anything Model [42], we provided bounding box prompts to obtain the object masks and point clouds. Grasp Pose Detection (GPD) [35] was used to generate the grasp poses



(a)



(b)

Fig. 12. Bar plots with error bars for each method across all participants for the complex scene user study. The plots for (a) objective measures and (b) subjective measures, where *= $p < 0.05$, **= $p < 0.01$, ***= $p < 0.001$, ****= $p < 0.0001$. $^{\vee}$ denotes that a lower score is better. The colour of the bars representing the control methods is consistent across all graphs.

from the object point cloud. In total, 22 goals were generated for the shared autonomy algorithms, as shown in Table IV. An AruCo marker [43] was placed to obtain the transformation between the arm and the generated grasp poses. The object names, grasp types and grasp poses were saved with respect to the AruCo marker and kept consistent across subjects.

To keep computation tractable, only the top 10 most likely goals were considered in the action space for *Policy*, as mentioned in Section III-B. Thus, the model had 22 actions, and a planning time of 0.1 seconds was used.

C. Procedure

We used a similar experimental procedure and metrics as outlined in Section V-C, with the differences explained below. We added a time limit of $100s$ to complete each task to prevent the experimental process from being overly lengthy. Participants had to complete the following three tasks (Fig. 11b) for each control method – *manual, HO* and *Policy*:

1) Grasp the bowl with a top horizontal grasp;
2) Grasp the can with a side horizontal grasp;
3) Grasp the drill with a side vertical grasp.

The tasks were carefully chosen to include tasks with different grasp types. The objects to grasp were also located at different locations in the scene, with the object in Task 1 being the closest to the starting position of the gripper and the object in Task 3 being the furthest away from the gripper. For the objects further away from the gripper, the gripper would have to pass by other objects, allowing us to evaluate whether our proposed method could reason what questions and whether to ask questions. The same objective and subjective measures in Section V-C were used for evaluation.

D. Results

The results in the complex scene user study show a similar trend to the results in the simple scene user study, with *Policy* performing at least as well or outperforming *manual* and *HO*. Similar to the previous study, for **task completion time**, **mode changes**, **total joystick input** and **opposing time**, the data reported is from the first attempt if the attempt was successful. If the first attempt failed but the second attempt was successful, the data reported is from the second attempt. If both attempts were unsuccessful, the data was excluded from the analysis of these measures.

1) Objective measures: A two-way repeated measures ANOVA was conducted to evaluate the effect of the task and assistance method on task completion time, mode changes and total joystick input. If the data violated the assumption of sphericity, a Greenhouse-Geisser correction was performed. Subsequently, if a significant main effect was found, we performed post hoc analysis with Holm-Bonferroni corrections to identify statistically different conditions. The results are shown in Fig. 12a.

A significant main effect was found for **task completion time** ($F(4, 112) = 2.466, p = 0.049$), but post hoc analysis did not show any significant differences. Thus, there is no

support for **H1**. This is likely due to the time limit of $100s$ placed in this set of experiments, which resulted in more failures.

For **success rates**, 33 failures occurred during the first attempt – 5 failures with *manual*, 21 failures with *HO* and 7 failures with *Policy*. If participants failed the first attempt, they were given a second attempt. For the second attempt, there were 10 failures for *HO* and 4 failures for *Policy*. Thus, there were 47 failures in total. Most failures occurred due to exceeding the time limit, with a total of 23 out of 47 failures attributed to this. There was only one case in *manual* control where the task failed due to exceeding the time limit, suggesting that the time limit was exceeded mainly due to opposition from the assistance provided. 19 failures in *HO* and 3 failures in *Policy* were attributed to exceeding the time limit. 20 failures were due to the object slipping after grasping, with 3 (all with *HO*) in Task 2 and 17 (2 with *manual*, 8 with *HO* and 7 with *Policy*) in Task 3. These failures were mainly because participants closed the gripper when the gripper was too far away from the object, resulting in an unstable grasp. The remaining failures were due to a collision with the environment (twice in *manual*, once in *Policy*) and a wrong grasp type (once in *HO* due to opposition from the assistance).

A significant main effect was found for **mode changes** ($F(4, 112) = 3.343, p = 0.013$) and **total joystick input** ($F(4, 112) = 4.868, p = 0.001$). Post hoc analysis found significant differences in **mode changes** between *manual* and *HO* ($p = 0.0329$) and between *manual* and *Policy* ($p = 0.0329$) for Task 3. For **total joystick input**, post hoc analysis revealed significant differences for most comparisons, with *manual* requiring significantly more total joystick input as compared to *HO* and *Policy*. Significant differences were observed between *HO* and *Policy* for Task 2 ($p < 0.01$) and Task 3 ($p < 0.001$), supporting **H2**.

A Wilcoxon signed-rank test was performed to compare the percentage opposing time and correct prediction rates for *HO* and *Policy*. For **percentage opposing time**, a statically significant difference was found for all tasks ($p < 0.001$), with *Policy* having a lower percentage opposing time. For **correct prediction rates**, a statistically significant difference was found for Tasks 1 ($p < 0.01$)and 3 ($p < 0.0001$), with *Policy* performing better. Thus, there is support for **H3**.

2) Subjective Measures: A Friedman's test was performed on the self-reported participant workload using the **NASA-TLX** survey. Significant differences were found for temporal, effort, frustration and performance measures, with *Policy* and *manual* outperforming *HO*. To compare the **usefulness** and **ease-of-use** of the shared autonomy algorithms, we performed a Wilcoxon signed-rank test, which showed significant differences for all measures between *HO* and *Policy*, suggesting that participants felt that *Policy* generally provided better assistance compared to *HO*. The results of these measures are shown in Fig. 12b.

80% of participants preferred *Policy* over *HO* (Fig. 13a), when asked which algorithm provided **better assistance**, most citing that *Policy* gave them more control compared to *HO*. When asked about their **preference** between all three control methods, 9 preferred *manual*, 2 preferred *HO* and 4 preferred *Policy*. Similar to the previous study, participants who selected *manual* mentioned that *manual* provided them with the greatest control.

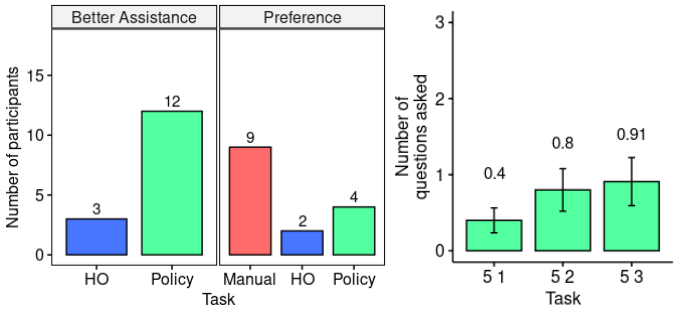


Fig. 13. (a) User preferences when asked which method provided better assistance and which method they preferred the most. (b) Number of questions asked by *Policy* for each task for the complex scene user study.

E. Discussion

The results obtained from this study were similar to the user study on a simple scene, where *Policy* outperformed *HO*. This suggests that *Policy* can scale to more complex scenes.

1) Number of Questions Asked: Even for a complex scene with more objects, the model only asked 0.4, 0.8 and 0.92 questions for each task (Fig. 13b), supporting **H5**. To provide more insights on the belief change and the questions asked, we selected Task 3 to analyse as it had the highest mean number of questions asked. Fig. 14 shows two runs by two different participants during the trial. The system only asked one question for the left plot in Fig. 14. Since the user answered "yes" to the question about *drill with side vertical* grasps, there was no more uncertainty, so no more questions were asked. For the right plot, two questions were asked. The first question was asked about *plate with top vertical* grasps, so the participant answered "no". Another question about *drill with side horizontal* grasps was asked subsequently, to which the participant also answered "no". With these questions, the system gathered sufficient information and was able to converge to the correct goal of *drill with side vertical* grasps. The results show that *Policy* can reason that when there are many goals, many questions will need to be asked to know the user's intent, which is less optimal than doing passive information gathering via move actions, such as move-to-goal, *hindsight optimization* and *no assist* actions.

2) Task Completion Time and Success Rates: Incorporating a time limit of 100 seconds was beneficial in this study, as it allowed us to more clearly distinguish cases where the assistance significantly opposed the user input. While the task may have been completed if given more time, it is more beneficial to categorize those instances as failures, given that the assistance impeded the task completion. Furthermore, by excluding failed attempts that exceeded the time limit from the analysis of completion time, our analysis of the objective measures becomes more representative of task performance.

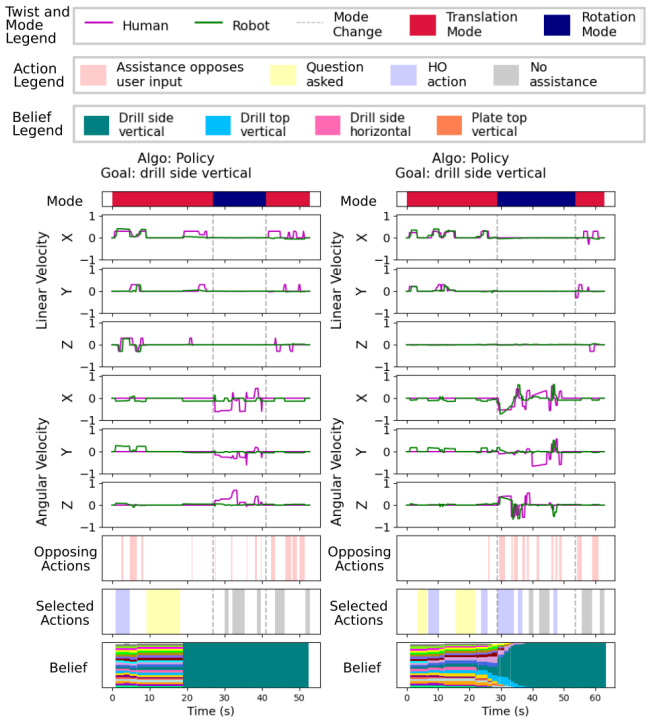


Fig. 14. Trajectories, actions selected and belief for *Policy* for Task 3 by two of the participants in the study. Only one question was asked for the run on the left as the user answered "yes" to the question. Two questions were asked for the run on the right. Even though the user answered "no" for both questions, the system gathered sufficient information to correctly predict the user's goal.

Given the average task completion time of approximately 50 seconds, the selected time limit of 100 seconds is adequate since it is double the average task completion time.

3) Perceived and Actual Effort: Surprisingly, although *manual* control demanded significantly more joystick inputs, participants generally did not perceive an elevated level of effort or physical demand associated with its use. Instead, they reported a significantly higher level of effort required for *HO* as compared to *manual* and comparable effort between *Policy* and *manual*. This contrast in reported effort could be due to the diminished control authority when using *HO*, which could have led to increased mental, physical and temporal demand and frustration levels. These factors likely contributed to the participants' perception of greater effort when using *HO*, highlighting that factors beyond total joystick input and mode changes affect users' perception of effort and usability. We note that this user study only involved participants with no upper limb impairments. Therefore, any potential additional effort due to an increase in joystick input may not have been noticeable to these participants. However, this observation may not apply in the same way to individuals with upper limb impairments.

4) Limitations due to Generated Grasp Poses: While participants preferred *Policy* over *HO*, they showed greater preference for *manual* over both shared autonomy algorithms in this study. Even though some participants acknowledged that "*manual* requires greater focus and effort", they preferred *manual* as it provided them "total control". This was especially so when fine-tuning the grasp pose, as both shared autonomy algorithms would generate policies to assist the user towards one of the generated grasp poses. The greater preference shown for *manual* in a more complex scenario could be due to users having more specific preferences to grasp an object to avoid the neighbouring objects. However, the user's desired grasp pose may not be in the list of grasp poses, leading to repetitive adjustments to reach their desired grasp pose and a loss of control. To enhance the perception of control, future work could consider planning for cases where the user's desired grasp pose is unknown or stop assistance when the gripper is within a certain distance threshold of an object.

5) Improvement in Second Attempt: The improvements in successive success rates suggest that users may adapt to the assistance provided. This was also observed as some participants changed their teleoperation strategy after observing that the robot failed to assist them in the first attempt. For example, some users first provided translational inputs to bring the gripper close to the object and provided rotational inputs to change the gripper orientation. If this strategy failed to complete the task in the first attempt, some users changed their strategy to rotate the gripper to the correct orientation at the start to compensate for incorrect intent recognition. For future work, a longitudinal study could be done to assess whether user preferences would change with experience.

VII. Conclusion

In this work, we presented a novel shared autonomy framework that allowed for active information-gathering and no assistance when uncertainty is high. By introducing information-gathering actions in the form of asking questions, our proposed method could disambiguate between goals better, which prevented assistance towards the wrong goal.

We conducted qualitative ablation experiments on the high-level actions included in our model. Without question-asking actions, the system was unable to predict the user's goal successfully in complex cases, resulting in greater user effort due to conflicting actions between the human and robot. By introducing a *no assist* action and allowing the system to arbitrate control, the performance improved. The introduction of a *hindsight optimization* action also improved the performance of the system by ensuring the system does not select random actions at the start.

A user study was also performed to evaluate our approach in the real world, showing that our proposed method resulted in better task completion time, fewer mode changes and joystick inputs compared to *HO* in complex scenarios. Even though our method has significantly reduced opposing actions compared to *HO*, some users still preferred *manual* due to greater control. One of the key reasons is that the user's desired grasp pose might not be in the list of generated poses. To address this issue, we can consider the history of a few time steps of human actions to determine opposing actions, which can allow the human to bring the gripper to any pose. Future work can also consider more informational questions and answers from the user via responses like "Yes, I want to grasp the object, but with a different grasp type".

Even though the question-asking actions are meant for the robot to gather information about the human's goal, many participants commented that they liked the questions asked as they can better "understand what the algorithm is trying to do". This shows the importance of transparency in what the system is doing, and simply by asking questions, users find that they can gain greater awareness of the system's behaviour. Increasing trust and transparency through other forms of statements could also be explored in the future.

REFERENCES

[1] V. L. Feigin, B. A. Stark, C. O. Johnson, G. A. Roth, C. Bisignano, G. G. Abady, M. Abbasifard, M. Abbasi-Kangevari, F. Abd-Allah, V. Abedi *et al.*, "Global, regional, and national burden of stroke and its risk factors, 1990–2019: a systematic analysis for the global burden of disease study 2019," *The Lancet Neurology*, vol. 20, no. 10, pp. 795–820, 2021.

[2] L. M. Haddad, P. Annamaraju, and T. J. Toney-Butler, "Nursing shortage," *StatPearls*, 1 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK493175/

[3] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa, "Assistive teleoperation of robot arms via automatic time-optimal mode switching," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2016, pp. 35–42.

[4] D. A. Abbink, T. Carlson, M. Mulder, J. C. F. de Winter, F. Aminravan, T. L. Gibo, and E. R. Boer, "A topology of shared control systems—finding common ground in diversity," *IEEE Trans. Human-Mach. Syst.*, vol. 48, no. 5, pp. 509–525, 2018.

[5] M. Young, C. Miller, Y. Bi, W. Chen, and B. D. Argall, "Formalized task characterization for human-robot autonomy allocation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6044–6050.

[6] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, 2018.

[7] K. Hauser, "Recognition, prediction, and planning for assisted teleoperation of freeform tasks," *Autonomous Robots*, vol. 35, no. 4, pp. 241–254, 2013.

[8] N. Mehr, R. Horowitz, and A. D. Dragan, "Inferring and assisting with constraints in shared autonomy," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 6689–6696.

[9] G. Quere, A. Hagengruber, M. Iskandar, S. Bustamante, D. Leidner, F. Stulp, and J. Vogel, "Shared control templates for assistive robotics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1956–1962.

[10] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[11] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," in *Proceedings of Robotics: Science and Systems*, 2015. [Online]. Available: http://www.roboticsproceedings.org/rss11/p32.pdf

[12] D. Gopinath, S. Jain, and B. D. Argall, "Human-in-the-loop optimization of shared autonomy in assistive robotics," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 247–254, 2016.

[13] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in shared autonomy," *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 294–302, 2017.

[14] C. Brooks and D. Szafir, "Balanced information gathering and goal-oriented actions in shared autonomy," *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 85–94, 2019.

[15] M. C. Fontaine and S. Nikolaidis, "A quality diversity approach to automatically generating human-robot interaction scenarios in shared autonomy," in *Proceedings of Robotics: Science and Systems*, July 2021. [Online]. Available: https://arxiv.org/abs/2012.04283

[16] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[17] S. Nikolaidis, M. Kwon, J. Forlizzi, and S. Srinivasa, "Planning with verbal communication for human-robot collaboration," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 7, no. 3, pp. 1–21, 2018.

[18] V. V. Unhelkar, S. Li, and J. A. Shah, "Decision-making for bidirectional communication in sequential human-robot collaborative tasks," in *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2020, pp. 329–341.

[19] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 362–370.

[20] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "Despot: Online pomdp planning with regularization," *Journal of Artificial Intelligence Research*, vol. 58, pp. 231–266, 2017.

[21] N. P. Garg, D. Hsu, and W. S. Lee, "Despot-alpha: Online pomdp planning with large state and observation spaces." in *Proceedings of Robotics: Science and Systems*, June 2019. [Online]. Available: http://www.roboticsproceedings.org/rss15/p06.pdf

[22] P. Cai, Y. Luo, D. Hsu, and W. S. Lee, "Hyp-despot: A hybrid parallel algorithm for online planning under uncertainty," *The International Journal of Robotics Research*, vol. 40, no. 2-3, pp. 558–573, 2021.

[23] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," *Advances in neural information processing systems*, vol. 23, 2010.

[24] Z. N. Sunberg and M. J. Kochenderfer, "Online algorithms for pomdps with continuous state, action, and observation spaces," in *Twenty-Eighth International Conference on Automated Planning and Scheduling*, 2018.

[25] S. Reddy, A. D. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," *arXiv preprint arXiv:1802.01744*, 2018.

[26] H. J. Jeon, D. P. Losey, and D. Sadigh, "Shared autonomy with learned latent actions," in *Proceedings of Robotics: Science and Systems*, 2020. [Online]. Available: http://www.roboticsproceedings.org/rss16/p011.pdf

[27] S. Karamcheti, M. Srivastava, P. Liang, and D. Sadigh, "Lila: Language-informed latent actions," *Conference on Robot Learning*, pp. 1379–1390, 2022.

[28] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011, pp. 1507–1514.

[29] S. Rosenthal and M. Veloso, "Modeling humans as observation providers using pomdps," in *2011 RO-MAN*. IEEE, 2011, pp. 53–58.

[30] M. Shridhar, D. Mittal, and D. Hsu, "Ingress: Interactive visual grounding of referring expressions," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 217–232, 2020.

[31] H. Zhang, Y. Lu, C. Yu, D. Hsu, X. La, and N. Zheng, "Invigorate: Interactive visual grounding and grasping in clutter," in *Proceedings of Robotics: Science and Systems*, 2021. [Online]. Available: http://www.roboticsproceedings.org/rss17/p020.pdf

[32] Y. Mo, H. Zhang, and T. Kong, "Towards open-world interactive disambiguation for robotic grasping," in *CoRL 2022 Workshop on Learning, Perception, and Abstraction for Long-Horizon Planning*, 2022.

[33] Y. Cui, S. Karamcheti, R. Palleti, N. Shivakumar, P. Liang, and D. Sadigh, "" no, to the right"–online language corrections for robotic manipulation via shared autonomy," *arXiv preprint arXiv:2301.02555*, 2023.

[34] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[35] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[36] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Human behavior modeling with maximum entropy inverse optimal control." in *AAAI spring symposium: human behavior modeling*, vol. 92, 2009.

[37] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE signal processing magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[38] L. Kish, *Survey sampling / Leslie Kish*. Chichester : Wiley New York, 1965.

[39] D. P. Losey, H. J. Jeon, M. Li, K. Srinivasan, A. Mandlekar, A. Garg, J. Bohg, and D. Sadigh, "Learning latent actions to control assistive robots," *Autonomous robots*, pp. 1–33, 2021.

[40] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Human Mental Workload*, ser. Advances in Psychology, P. A. Hancock and N. Meshkati, Eds. North-Holland, 1988, vol. 52, pp. 139–183. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166411508623869

[41] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference,*

Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, 2014, pp. 740–755.

[42] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[43] F. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, 06 2018.

J-Anne Yow received a bachelor's degree in Mechanical Engineering in 2021 from Nanyang Technological University (NTU), Singapore. She is currently working towards a Ph.D. degree at the Rehabilitation Research Institute of Singapore, NTU.

Her research interests include human-robot interaction, natural language processing and robot learning.

Neha Priyadarshini Garg received her Bachelors of Technology in Computer Science from Indian Institute of Technology Delhi, India in 2006 and Masters in Computer Science from Ecole Polytechnique Federal de Lausanne (EPFL), Switzerland in 2008. Then, she worked as a Software Engineer at a computer vision startup Kooaba in Zurich, and a mobile software development company Affle in Singapore. She received her PhD degree from National University of Singapore, School of Computing under NUS Graduate School's Integrative Sciences and Engineering Programme (ISEP) in 2020 during which she worked on autonomous grasping under uncertainty using POMDPs. She is currently a Research Fellow at the Rehabilitation Research Institute of Singapore, where she is working on human-robot interaction algorithms for assistive robots like robotic wheelchairs and robotic arms. Her research interests include human-robot interaction and planning under uncertainty.

Ang Wei Tech graduated with a PhD degree in Robotics from the Robotics Institute, Carnegie Mellon University, USA in 2004, and M.Eng. and B.Eng. degrees in Mechanical Engineering from NTU in 1999 and 1997 respectively. He is currently an Associate Professor at the School of Mechanical & Aerospace Engineering, NTU, and concurrently as the Executive Director of the Rehabilitation Research Institute of Singapore, a joint collaboration by Nanyang Technological University (NTU), Agency for Science, Technology and Research (A*STAR) and National Healthcare Group (NHG).

Prof Ang's research focuses on robotics technology for biomedical applications, which include surgery, cell micromanipulation, rehabilitation, and assistive technology. His work has been well funded, published and cited, and resulted in several inventions licensed to the industry and spin-off companies.