Landmark-Based Goal Recognition for Shared Autonomy: A Framework for Enhanced Teleoperation

Guillaume Lorthioir[†*], Mehdi Benallegue[†], Rafael Cisneros-Limón[†], Ixchel G. Ramírez-Alpizar[‡]

***Abstract*— Shared autonomy is the future of teleoperation as it reduces the teleoperator's burden, enhances capabilities, and improves embodiment by offering seamless control of the robot. However, it remains rarely used, particularly with humanoid robots, as it faces numerous challenges. In this work, we introduce an innovative shared autonomy framework suitable for a wide range of robots, which we tested on a humanoid robot. This framework leverages Bayesian filtering over a Hidden Markov Model (HMM) to perform goal recognition, employing a landmark-based heuristic that minimizes computational demands while computing observation likelihoods without prior knowledge or a cost function. Once the teleoperator's goal is identified, the robot assists according to its confidence level in the goal prediction. Assistance is provided by guiding the robot's end-effector to reach a specified target position and orientation. In experiments with a diverse group of 10 teleoperators, conducted with video transmission delay, we achieved high accuracy in goal prediction and demonstrated significantly faster teleoperation time with shared autonomy.**

I. INTRODUCTION

Robots are not yet fully autonomous, and in many cases, a human needs to teleoperate them, such as doctors during surgery, interventions in hazardous places, or even to embody human presence in remote locations, as seen in the ANA Avatar XPRIZE competition [1]. Unfortunately, teleoperation in robotics is not always straightforward; several factors can impede the experience, such as communication delays, noises, obstruction of the field of view, a complicated interface, or the experience level of the teleoperator. One approach to enhance teleoperation is through the utilization of shared autonomy [1]. This method involves the robot remaining partially autonomous, thereby assisting the teleoperator. Typically, the robot attempts to discern the teleoperator's goal to facilitate its achievement. For example, a robotic arm capable of identifying the object the teleoperator intends to grasp automatically positions its gripper appropriately for retrieval. Naturally, each instance of shared autonomy has specific requirements dependent on the robot and the teleoperator. In most cases, however, the problem can be subdivided into two main components: Goal Recognition and Decision-making. Goal Recognition involves inferring the intentions of an observed agent, where observations may be actions or facts and can be subject to noise. In the context of teleoperation, the agent is the teleoperator, and we often possess comprehensive knowledge about their actions through the robot. In addressing the second subproblem, Decision-making, we must determine when and how to assist the teleoperator with the robot. Given that our goal prediction may not always be accurate, and each user may have individual preferences regarding the level of assistance provided by the robot, decision-making becomes a crucial aspect of shared autonomy.

We present a shared autonomy system for humanoid teleoperation, leveraging a landmark-based heuristic for online goal recognition. Unlike previous approaches focused on robotic arms or wheelchairs [2], [3], our method enables adaptive and scalable goal inference by incorporating semantic planning and reducing reliance on predefined trajectory-based models. The use of Bayesian filtering with a landmark-based representation allows seamless integration of new goals without retraining, making it well-suited for real-time applications. While robotic arms benefit significantly from shared autonomy due to their clumsy game controller-based operation, humanoid robots offer greater dexterity and suitability for human-centered environments. Although their many DoF add complexity, VR-based control enables simultaneous DoF management. Our results demonstrate that shared autonomy still provides substantial benefits in this context. Additionally, humanoids' human-like appearance makes them more approachable, and their two hands enhance manipulation capabilities. However, our framework is not limited to humanoids and can be easily adapted to other robotic platforms. Our goal recognition methodology is computationally efficient, avoiding the need to invoke a planner for each potential goal at every inference step, a limitation in many shared autonomy frameworks [4], [2]. Instead, the system initializes goal-specific landmarks (key observations or actions) at the start of a task, significantly accelerating inference without compromising accuracy. Bayesian filtering, integrated with a Hidden Markov Model (HMM), continuously updates the probability distribution of potential user goals. An entropy-based metric determines when the robot should intervene, ensuring assistance is provided only when necessary, enhancing both efficiency and user experience.

This paper follows with Section II, which explores related work in goal recognition and shared autonomy. Next, Section III introduces the goal recognition algorithm developed for our shared autonomy framework, and Section IV outlines the decision model used to determine when and how to assist the teleoperator. Section V explains the full integration of our shared autonomy framework, detailing how the goal recognition and decision model come together. Subsequently,

*gui.lorthioir@gmail.com

[†]CNRS-AIST JRL (Joint Robotics Laboratory), IRL3218, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan

[‡]Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

[1]https://www.xprize.org/prizes/avatar

Section VI documents the experiments conducted to evaluate our framework, while Section VII provides conclusions and discussions concerning the contributions made by this paper.

II. RELATED WORK

A. Goal recognition

Goal recognition infers a user's intended goal from observed actions and environmental cues. Early methods relied on elimination-based approaches [5], [6], but they lacked accuracy or were too computationally expensive for real-time use. Later, heuristic-based methods improved efficiency by reducing planning complexity. E-Martín *et al.* [7] used cost estimates and plan graphs, while Vered and Kaminka [8] extended goal recognition to continuous spaces. Pereira *et al.* [9], [10] introduced landmark-based heuristics, achieving state-of-the-art results for real-time applications.

B. Shared autonomy

Shared autonomy balances control between humans and robots by predicting user goals and providing adaptive assistance. Dragan and Srinivasa [11], [12] introduced policy blending, where the robot adjusts its assistance based on confidence in goal prediction. Javdani *et al.* [4], [13] later proposed a more autonomous approach using a Partially Observable Markov Decision Process (POMDP) to infer the most likely goal, improving efficiency but reducing user control, which some participants found undesirable. Early methods primarily relied on end-effector trajectories for goal prediction. To enhance recognition, Aronson *et al.* [14] explored eye tracking as an additional input, demonstrating its usefulness. Jain and Argall [2] extended goal recognition with a POMDP-Bayesian framework, enabling the integration of new observations. However, their method requires goal-specific planning at every inference step, raising concerns about scalability for larger goal sets and long-term tasks.

Our work builds on these studies by introducing a computationally efficient, landmark-based heuristic for goal recognition. Unlike methods that require trajectory modeling or goal-specific planning, our approach generalizes across different robots using Bayesian filtering within an HMM. This framework allows new goals to be incorporated seamlessly without retraining, enhancing adaptability for real-time applications. We demonstrate these advantages in the context of humanoid teleoperation, an area that remains underexplored despite the growing adoption of humanoid robots for human-centered tasks.

III. GOAL RECOGNITION MODEL

A. Problem Formulation

We observe the user controlling the robot to perform various tasks. The user has a discrete set of goals $G = g_1, g_2, ..., g_n$ of size n that they aim to achieve, and we observe their actions (inputs to the controllers, eye tracking, robot motions, etc.). We denote $\Theta_t = \theta_t^1, \theta_t^2, ..., \theta_t^k$ as the vector of observations obtained at time t, where k is not constant. At times t and $t-1$, we may have a different number of observations. Through these observations, our objective is to infer the current goal g^* of the user in real-time. We assume our observations are comprehensive as they directly come from the feedback of our teleoperating system. However, the user is acting based on their own volition and may change their intended goal or even be undecided during the execution of their task.

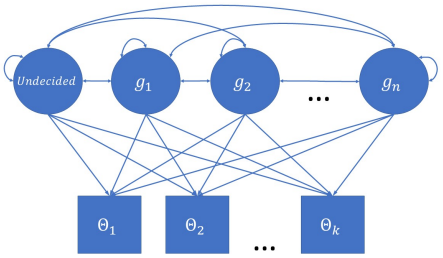

Fig. 1. This HMM diagram represents our goal recognition problem. At the top are the different states in G and their transitions, which influence the bottom layer, the set of observations O.

Our approach to goal recognition is inspired by [2]. We formulate the goal recognition problem as Bayesian filtering over an HMM. The hidden states of this HMM correspond to the different goals that the user can achieve in G, along with a state `Undecided` representing the user's current lack of commitment to any specific goal. For clarity, we consider `Undecided` $\in G$. The observations O of the HMM constitute the set of all possible observation vectors, such that $\forall t, \Theta_t \in O$. Figure 1 illustrates the HMM described, where the user navigates between the upper states of the HMM, G, and their position directly influences the likelihood of the observations (depicted in the bottom part of the diagram). We can model a probability distribution over G corresponding to the uncertainty over the candidate goals using the Forward Algorithm. We consider first the case of a single observation at each time step, where $\Theta_t = \theta_t$ is a vector of dimension one. We use the colon notation $\theta_{0:t}$ for the sequence of observation $\theta_0, ..., \theta_t$; then, we define $b_t(g_t)$ as the belief of the user following the goal $g \in G$ at time t, with:

$$b_t(g_t) = P(g_t, \theta_{0:t}) = \sum_{g_{t-1} \in G} P(g_t, g_{t-1}, \theta_{0:t}) \quad (1)$$

Eq. (1) is obtained by applying the law of total probability, then, using the chain rule it becomes:

$$b_t(g_t) = \sum_{g_{t-1} \in G} P(\theta_t | g_t, g_{t-1}, \theta_{0:t-1}) * P(g_t | g_{t-1}, \theta_{0:t-1}) P(g_{t-1}, \theta_{0:t-1}) \quad (2)$$

And lastly, since θ_t is conditionally independent of everything but g_t, and g_t of everything but g_{t-1}, Eq. (2) simplifies to Eq. (3).

$$b_t(g_t) = P(\theta_t | g_t) \sum_{g_{t-1} \in G} P(g_t | g_{t-1}) b_{t-1}(g_{t-1}) \quad (3)$$

Eq. (3) is employed to compute $b_t(g_t)$ when there is only one observation at each time step. However, considering multiple

observations is desirable as additional data typically leads to a more accurate prediction of goals. We now examine the scenario where $\Theta_t = \theta_t^1, \theta_t^2, ..., \theta_t^k$ has a dimensionality $k > 1$. We assume that the various observations $\theta_t^1, \theta_t^2, ..., \theta_t^k$ are conditionally independent of each other given a goal g. Thus, Eq. (3) can be expressed as:

$$b_t(g_t) = (\prod_{\theta_t \in \Theta_t} P(\theta_t|g_t)) \sum_{g_{t-1} \in G} P(g_t|g_{t-1}) b_{t-1}(g_{t-1}) \quad (4)$$

Evaluating this expression for all $g \in G$ provides us with the estimated distribution b_t of our belief regarding the user's goal over G at time t. Since our objective is to infer the most likely goal g_t^* the user is striving to achieve, we compute:

$$g_t^* = arg\max_{g_t \in G} b_t(g_t) \quad (5)$$

Utilizing the recursive Algorithm 1 (a slightly modified version of the Forward Algorithm that accommodates multiple observations), we can determine g_t^* at any time t without necessitating exponential computation time, enabling real-time goal inference. Moreover, Bayesian inference methods have demonstrated effectiveness in human goal and intent recognition, exhibiting robustness to noisy data [15], [16].

Algorithm 1: Goal Recognition Algorithm

input : b_0, G
output: g_t^*

while *a goal is not achieved* **do**
  get Θ_t
  for g *in* G **do**
    $b_t(g_t) \leftarrow \prod_{\theta_t \in \Theta_t} P(\theta_t|g_t) \sum_{g_{t-1} \in G} P(g_t|g_{t-1}) b_{t-1}(g_{t-1})$
  end
  normalize b_t
  $g_t^* \leftarrow arg\max_{g_t \in G} b_t(g_t)$
end
return g_t^*

Algorithm 1 takes G and b_0 as input, where b_0 represents the initial distribution of our beliefs over G. We initialize it as a uniform distribution, such that $\forall g \in G, b_0(g_0) = \frac{1}{n}$. Subsequently, as long as the user has not yet achieved a goal, we continuously update the beliefs over G and ascertain the most probable goal at each time t. Upon the user accomplishing a goal, we simply reset and adjust b_0 and G accordingly, and proceed to execute the algorithm for the subsequent inference task.

B. Parameters Estimation

The challenge now lies in the uncertainty surrounding the precise values of $P(\theta_t|g_t)$ and $P(g_t|g_{t-1})$, which can only be estimated. The optimal estimation would involve conducting numerous trials to gather user data, followed by statistical analysis to derive estimators. However, this approach is excessively time-consuming and must be repeated for each potential goal requiring prediction. Jain and Argall [2] proposed a heuristic based on the distance between the end-effector and the goal position to approximate $P(\theta_t|g_t)$. According to this heuristic, the closer the end-effector is to the goal, the higher the probability. Nonetheless, this method encounters challenges when multiple goals are in close proximity to each other or lie along the trajectory of others. To address these limitations, an alternative heuristic based on a cost function was explored, comparing user actions to an optimal plan for achieving each goal. The goal associated with the lowest cost according to this function is considered the most likely. A similar approach was also utilized by [12]. However, obtaining optimal plans can impose significant computational burdens. Even when pseudo-optimal planners are employed, recent studies in goal recognition, such as [10], suggest that utilizing a Landmark-based heuristic could offer improved efficiency. During goal pursuit, a landmark refers to a fact or action that must be true or performed to achieve the goal. We adopt the principle of Landmark Uniqueness, which involves assessing the appearance of a landmark across different goals. For a given landmark L, its uniqueness U_L is defined as follows:

$$U_L = \frac{1}{\sum_{\mathcal{L}_g \in \mathcal{L}_G} |\{L|L \in \mathcal{L}_g\}|} \quad (6)$$

Here, $\mathcal{L}_g$ represents the set of landmarks associated with goal g, and $\mathcal{L}_G$ denotes the set comprising all landmarks designated for every goal in G. In essence, a landmark's uniqueness is inversely proportional to the number of goals with which it is associated. We use this uniqueness metric to approximate $P(\theta_t|g_t)$.

$$P(\theta_t|g_t) = \begin{cases} max(\beta e^{U_L - 1}, \frac{1-\beta}{|G|}) & \text{if } L \in \mathcal{L}_g \\ \frac{1-\beta}{|G|} & \text{else} \end{cases} \quad (7)$$

Here, $\beta \in [0, 1[$ is a parameter used to adjust the strength of the landmark weights in computing $b_t(g_t)$ in Eq. (4). Eq. (7) also ensures that $P(\theta_t|g_t)$ does not equal zero, which would be problematic as it would result in $b_{t+1}(g_{t+1}) = b_t(g_t) = 0$ and so forth for all subsequent indices due to the recursive nature.

Eq. (7) also ensures that observing L if $L \in \mathcal{L}_g$ has a positive impact on $P(\theta_t|g_t)$ (in the case where β is too low), it would not make sense otherwise.

Utilizing such a heuristic enables us to circumvent the need for a planner, as we solely require a list of necessary steps to achieve the goal (i.e., the landmarks), which does not need to be sorted or complete. However, a more comprehensive list tends to yield better performance.

To compute $P(g_t|g_{t-1})$, representing the probability of the user changing its goal, we utilize the same formula as [2]:

$$P(g_t|g_{t-1}) = \begin{cases} 0.9 & \text{if } g_t = g_{t-1} \\ \frac{0.1}{|G|-1} & \text{else} \end{cases} \quad (8)$$

IV. DECISION MODEL

Determining the appropriate level and timing of robotic assistance is a complex challenge. First, our goal recognition

model is inherently fallible. As noted by Meneguzzi and Pereira [17], achieving perfect online goal prediction remains an open problem. It is therefore crucial to acknowledge the possibility of errors, as incorrect predictions may lead to inappropriate assistance and ultimately hinder rather than help teleoperation. To manage this, we quantify the confidence of each prediction. Dragan and Srinivasa [12] proposed several methods for computing confidence. We initially adopt the approach of measuring the gap between the top two predicted goals. High confidence is assumed when the most probable goal significantly outweighs the others, as also used in [2], and defined by:

$$\gamma = b_t(g_t^*) - arg \max_{g_t \in G \setminus g_t^*} b_t(g_t) \tag{9}$$

However, this approach has limitations. When the two most probable goals have similar target positions, their estimated probabilities are often close. As a result, according to Eq. (9), γ tends toward zero, which suppresses assistance from the shared autonomy framework. While continued teleoperator actions eventually differentiate the probabilities and increase γ, assistance is delayed during this period. To mitigate this issue, we redefine confidence using the entropy of the probability distribution b_t, as also proposed in [12], and denoted H:

$$H = -\sum_{g_t \in G} b_t(g_t) log(b_t(g_t)) \tag{10}$$

Entropy is maximized when the distribution is uniform. Hence, we define $H_{max} = -\log(1/n)$ as the maximum attainable value for H. Accordingly, our confidence metric (γ) is redefined as:

$$\gamma = 1 - \frac{H}{H_{max}} \tag{11}$$

In Eq. (11), as b_t approaches a uniform distribution, γ tends towards 0, indicating minimal confidence in our prediction, as no goal appears distinctly more probable than the others. Conversely, when b_t assigns high probabilities to only a few goals, resulting in low entropy, γ approaches 1, signifying high confidence that the goal recognition algorithm has accurately identified a small set of probable goals.

The second challenge involves determining the extent of autonomy afforded to the user, a decision heavily influenced by the user's comfort level with the robot's autonomy. We employ policy-blending, a widely utilized technique in shared autonomy [12], [2], as detailed in Section II-B. This approach involves combining the desired trajectory dictated by the user (u_h) with the robot's preferred trajectory (u_r) to derive the blended policy (u_b), expressed as:

$$u_b = u_h(1-\alpha) + u_r\alpha \tag{12}$$

Here, $\alpha \in [0, 1]$ represents a parameter governing the balance between the human and robot influences in the blended policy. The determination of α hinges on the confidence metric (γ) discussed earlier. Prior studies such as [12], [2] typically model α as a linear function of γ. They reveal through subjective evaluations that users generally disfavor incorrect assistance from the robot (resulting from erroneous prediction of g^*), yet are receptive to the robot taking the lead when predictions are accurate. Building upon these insights, we opt to compute α as a linear function of γ, providing minimal assistance to the user when confidence is low to avoid unnecessary disruptions, and progressively increasing assistance as confidence rises. The computation of α is defined in Eq. (13):

$$\alpha = \begin{cases} 0 & \text{if } \gamma \leq \delta_1 \\ \gamma & \text{if } \delta_1 < \gamma \leq \delta_2 \\ \delta_2 & \text{if } \delta_2 < \gamma \end{cases} \tag{13}$$

Here, δ_1 and δ_2, both within the range $]0, 1]$, denote the threshold for confidence above which assistance is enabled and the upper limit for the weight of the robot policy (u_r) in u_b, respectively. These parameters can be customized for each user based on their comfort level with robot assistance.

A problem that could happen would be having the user stuck in a loop of the robot trying to achieve a goal inferred with very high confidence and the user trying to cancel the motion because it is actually not the goal they are trying to achieve. When we introduced the HMM used for our goal recognition problem we mentioned the use of a state `Undecided`. This state is there to solve this issue. A strong landmark of the `Undecided` state is when the user attempts to move the robot's hand away from the current most likely goal. When this is observed, the belief in the `Undecided` state will increase drastically since this landmark has a uniqueness $U_L = 1$, thereby stopping the robot's assistance toward the incorrect goal. When the most likely goal inferred is `Undecided`, the robot does not provide any assistance. It is also good to represent the fact that the user might not be trying to achieve any goal currently, and thus there is no need to provide any assistance.

V. COMPLETE FRAMEWORK

Now that we have described the two main components of our shared autonomy framework, we proceed to explain its operation in its entirety.

We begin by determining the uniqueness of all landmarks. While the user has not yet achieved a goal, Algorithm 2 is executed. This algorithm receives new observations Θ_t at a fixed frequency, processes them to compute the most likely goal g_t^*, calculates the confidence metrics γ and α, and determines the policy u_b to move the robot's end-effector toward the desired position and orientation. This loop continues until a goal is successfully achieved. Upon goal completion, b_0 and G are updated, and the algorithm restarts for the next goal. Since the robot has two end-effectors (hands), we assign assistance based on proximity: the end-effector closest to the predicted goal position is selected. If this arm is not actively controlled by the user (i.e., not moving), assistance is instead applied to the currently active arm. In cases where two goals have exactly the same predicted probability, one is selected at random. This ambiguity typically resolves quickly as continued teleoperator motion causes the probabilities to diverge and converge toward the actual goal.

Algorithm 2: Shared autonomy Algorithm

input : b_0, G
output: g_t^*

$\forall L \in \mathcal{L}_\mathcal{G}$ compute U_L
while *a goal is not achieved* **do**
  get Θ_t
  for g *in* G **do**
    $b_t(g_t) \leftarrow \prod_{\theta_t \in \Theta_t} P(\theta_t|g_t) \sum_{g_{t-1} \in G} P(g_t|g_{t-1}) b_{t-1}(g_{t-1})$
  end
  normalize b_t
  $g_t^* \leftarrow arg\,max_{g_t \in G}\, b_t(g_t)$
  $\gamma \leftarrow 1 - \frac{H}{H_{max}}$
  compute α
  $u_b \leftarrow u_h(1-\alpha) + u_r\alpha$
  apply u_b
end
return $b_t(G)$

The robot's assistance is implemented as an absolute position control task for the end-effector, guiding it toward a target pose. Currently, no obstacle avoidance is integrated. However, the robot's motions are deliberately slow, allowing the teleoperator to retain effective control, as the actual command sent to the controller is a blend of the robot's target and the teleoperator's input.

The simplest way to verify goal achievement is to check if the robot's hands have reached the specified position and orientation. This verification process is straightforward for basic goals such as "grab object X," where it involves checking if the hand is in contact with the object. However, for more complex goals like "pour X into Y," rule-based reasoning is necessary to determine goal achievement, which is currently hard-coded. This approach is consistent with previous works such as [12], [4], [2], which predominantly focus on goals like "grab object X" in their experiments.

VI. EXPERIMENT

A. Settings

In our experiments, we used MuJoCo [18], a free and open-source physics engine recognized for its high performance in simulating multi-contact forces during robotic grasping tasks. Fig. 2 shows the MuJoCo scene used for these experiments. We simulated the HRP-4CR robot [19], controlled using mc_rtc[2] with a MuJoCo interface called mc_mujoco [20]. This controller is a simplified version of the one developed for the ANA Avatar XPRIZE contest [3]. The simplification involved disabling the walking interface and streamlining the user interface to improve the repeatability of the experiment. To achieve a realistic simulation, we ensured

[2]https://jrl-umi3218.github.io/mc_rtc/index.html
[3]https://www.xprize.org/prizes/avatar

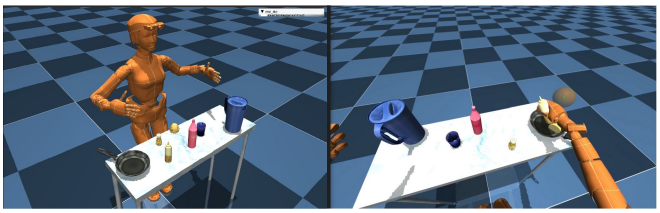

Fig. 2. Simulation environment for the experiment and Teleoperator's view while attempting to achieve the goal *pour_sauce* (left-eye perspective shown).

adherence to joint, kinematic, and dynamic constraints. However, to prevent the robot from falling due to environmental interactions (particularly when in contact with the table) the robot's waist was fixed in the scene. The challenge of accurately estimating the forces applied by a teleoperated robot on its surroundings has been partially explored in [19]. This remains a complex problem, as teleoperators often struggle to prevent the robot from losing balance during manipulation tasks. Grasping was also simplified: the robot's hands are equipped with virtual suction grippers in their palms, and gravity is temporarily disabled for any object held by the robot. This approach prevents issues like objects flying away or overreacting to collision forces from the robot's fingers, an issue that arises due to MuJoCo's limitations in accurately computing complex collisions. Addressing this issue with MuJoCo would improve robot manipulation capabilities in simulation, but unfortunately, resolving it is beyond the scope of this paper. We opted not to conduct experiments on the real robot due to the significant engineering requirements for real-time object detection, 6D pose estimation, and grasp-pose estimation. The development of these tools is beyond the scope of this paper, which focuses on demonstrating the increased effectiveness of teleoperation with shared autonomy. The teleoperator is tasked with cooking a potato and preparing a drink, which can be completed in any order. Cooking the potato requires placing it in the frypan and then pouring salt and sauce (from the yellow bottle) over it. Preparing a drink involves pouring from the red bottle and the pitcher into a glass. Each pouring action requires positioning the robot's hand at a precise target and holding still for 2s. After this, a new target appears, and the process is repeated three times. The right part of Fig.2 shows the teleoperator's view while controlling the robot. Here, the teleoperator is attempting the *pour_sauce* goal, with one of the targets displayed. The target appears as a small green ball within a larger orange sphere, guiding the teleoperator to position the robot's hand accurately. These precise pouring actions are designed to simulate fine motor tasks. A script automatically resets any objects that fall to the ground, returning them to their initial positions on the table after a five-second delay. This brief time penalty approximates the time it would take for a person to pick up and replace the object on the table.

The landmarks observed during the experiment include: the robot's hand moving closer to an object or goal position (i.e., the 3D distance between the hand and the position

decreases compared to the previous time step), the robot's hand aligning with an object (i.e., the orientation relative to the grasping position of the object improves compared to the previous time step), the closest object to the robot's hand, the user looking at an object for more than 1 second (as we implemented a gaze-tracking feature in MuJoCo), and the different goals that have already been achieved. Landmarks such as getting closer to, aligning with, and looking at an object, as well as identifying the closest object, have low uniqueness ($U_L \simeq 1/|G|$) since they are associated with most goals. In contrast, when a specific goal is achieved (such as grabbing the pitcher), the uniqueness of this landmark is significantly higher, as grabbing a pitcher is uniquely related to the goal pour pitcher. Additionally, two landmarks are unique to the `Undecided` state. The first, as discussed earlier in Section III, occurs when the user moves the hand away from the most likely goal (i.e., the 3D distance to the target position increases compared to the previous time step). The second corresponds to the absence of observable actions, which is interpreted as the user being in the `Undecided` state.

Eye tracking in MuJoCo is implemented by attaching a virtual cone to the robot's camera. The position and orientation of this cone dynamically adjust based on a directional vector representing the user's gaze direction. This gaze vector is obtained through the Vive eye-tracking software from the Vive Pro Eye HMD (SRanipal)[4]. When this virtual cone overlaps with an object for more than 1 second, we consider the user to be focusing their gaze on that object. While studies have shown that a human fixation typically lasts around 250 ms, using such a short duration for intention prediction through gaze is not recommended. Instead, a time window between 0.5 and 2 seconds yields the best results [21]. The cone serves purely as a projection, has no physical interaction with the environment, and remains invisible to the teleoperator.

The observation vector at time t consists of all the landmarks observed at that time. For instance, it could be:

$$\Theta_t = \begin{array}{l}\{moving\_closer(Salt), moving\_closer(Sauce),\\ looking\_at(Salt), closest\_object(Salt)\}\end{array}$$

where each element of Θ_t represents an observation made at time t (a landmark). In this case, the end effector is moving closer to both the salt and the sauce, the teleoperator is looking at the salt, and the salt is the closest object to the end effector. Basic landmarks can be observed or not at any time step. However, for more complex ones that involve an action previously performed by the teleoperator, such as pouring salt into the frypan, once the action has been completed, it will always be included in the observation vector Θ_t. To account for this, we introduced a memory parameter that gradually decreases the weight of such landmarks over time, making their influence negligible after 20s. This models the idea that if the teleoperator does not complete the next task associated with the action within this timeframe, they have likely changed their mind and intend to pursue a different goal.

[4]https://www.vive.com/sea/product/vive-pro-eye/overview/

TABLE I

PARTICIPANTS' SELF-ASSESSED EXPERTISE IN VR AND TELEOPERATION ON A SCALE FROM 1 (NO EXPERIENCE) TO 10 (EXPERT).

Participant	1	2	3	4	5	6	7	8	9	10
VR expertise	1	9	3	2	7	4	1	5	6	6
Teleop expertise	1	7	6	2	5	2	3	2	6	6

For the different parameters introduced in previous equations, we used $\beta = 0.75$, $\delta_1 = 0.2$, and $\delta_2 = 0.75$, so that α remains bounded between these values when assistance is enabled. This choice ensures that the teleoperator retains a minimum level of control over the robot. The beliefs over the goals, and thus α, are updated at a frequency of 4Hz, with observations collected at the same frequency just before each update. However, for α, we observed that frequent updates can lead to abrupt robot movements if α decreases suddenly (often due to inaccuracies in eye tracking, as this system is not flawless). Therefore, we decided to send the averaged value of α over the last 2s at each update, which prevents sharp variations in short time intervals. These parameter values were empirically chosen through trial and error, as they yielded the best results.

B. Protocol

Experiments were conducted with 10 participants (5 men and 5 women) ranging in age from 20 to 28 years, a similar number of participants to previous studies [4], [13] that was sufficient to perform a relevant statistical test. They had varying levels of expertise in teleoperation and VR, though most were beginners in these areas. Tab. I details the participants' expertise.

Each participant performed the manipulation task six times: three with shared autonomy and three without. The sequence order was randomized as follows: consider the pair $(0, 1)$, where 1 indicates using shared autonomy and 0 indicates not using it. The sequence for each participant was defined by three such pairs, which could randomly be $(0, 1)$ or $(1, 0)$. For example, the first participant received the sequence $(1, 0, 0, 1, 1, 0)$, ensuring that the same framework was not used more than twice consecutively and did not start or end with the same framework twice in a row. We evaluated both their task completion time and the number of times they caused an object to fall from the table while attempting to grasp it. If a participant's manipulation time exceeded 360s, they were stopped, and the remaining time was extrapolated based on the goals achieved during the task.

The video transmission to the VR headset, through which participants controlled the robot, had a delay of 1.5s. Such delays are common in teleoperation, especially in long-distance scenarios. For instance, teleoperating from the other side of the world or from Earth to the Moon typically results in delays of 1 to 2s [22]. We selected 1.5s as a middle ground, simulating a near worst-case scenario. This level of delay has not been extensively studied in shared autonomy literature, making it a valuable test case for our experiments.

Our experiment aims to validate the following hypothesis:

- The shared autonomy framework enables faster manipulation than direct teleoperation in the presence of video delay.

We applied the Friedman test due to the repeated measures design, small sample size, and non-normal data.

C. Results

Fig.3 presents the estimated goal probabilities up to 15s before achievement (defined as reaching the first of three targets for pouring goals). For most grasping goals, the probability exceeds 0.5 approximately 6s before completion. An exception is grabbing the potato, often the teleoperator's first action, where the heuristic has access only to gaze and end-effector distance data. In contrast, grabbing the pitcher—typically the penultimate action—benefits from more achieved landmarks. A similar pattern is observed for pouring goals: their distinctive landmarks maintain high probability estimates well before goal completion. However, 10s prior, probabilities remain low since most pouring actions complete in under 10s after object acquisition, particularly with shared autonomy assistance. At that time, the teleoperator is often still focused on grasping the required object. Goal prediction was performed at 4Hz, which proved sufficient while preserving computational resources for other robotic processes. These results suggest our heuristic achieves a strong balance of performance and efficiency.

Comparing Tab.I and Tab.II, we find that manipulation time does not correlate directly with self-reported expertise. For example, Participants 7 and 8 achieved the fastest times despite modest ratings (expertise: 2 and 3; VR experience: 4 and 1), whereas Participant 2, with the highest self-rated expertise, did not perform especially well. This discrepancy may stem from the subjective nature of self-assessment. As shown in Tab.II, all participants performed tasks faster with shared autonomy. On average, manipulation time decreased from 318.58s to 229.97s, a 38.5% improvement. A Friedman test across all trials (not averages) confirms this effect as statistically significant ($p < .0016$), supporting our hypothesis that shared autonomy enhances task efficiency under video delay conditions. Standard deviation analysis does not indicate whether performance was more consistent with shared autonomy. Furthermore, shared autonomy did not significantly affect failure rates ($p > .25$).

A supplementary video included with this paper demonstrates a teleoperator completing tasks both with and without shared autonomy. A higher-quality version is available on YouTube[5].

VII. DISCUSSION

We presented a shared autonomy framework based on a goal recognition landmark heuristic that requires only a list of landmarks and minimal computation, yet provides accurate predictions. This framework was tested on the humanoid robot HRP-4CR under video transmission delays.

[5]https://youtu.be/QF8uWj5-Ks4

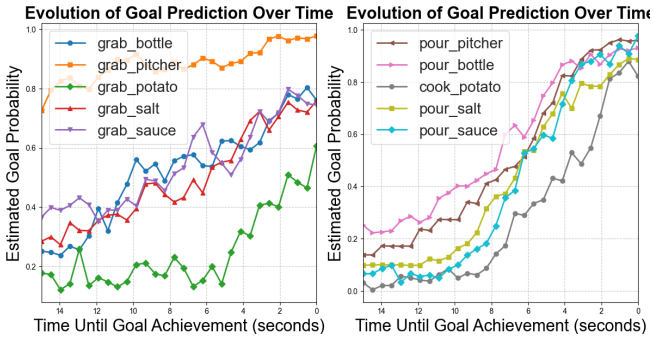

Fig. 3. Average estimated probability for each goal up to 15s before achievement, with basic goals displayed at the top and more complex goals at the bottom. Calculated across 30 samples using shared autonomy.

As expected, shared autonomy significantly improved performance in complex manipulation tasks during teleoperation, reducing manipulation time by approximately 38.5%. We believe this approach could greatly reduce teleoperator fatigue during longer tasks. Improved teleoperation under delayed conditions is particularly relevant to real-world scenarios, where delays are common and robots must remain functional without making critical errors. Interestingly, the benefits of shared autonomy appeared independent of the teleoperator's self-assessed expertise, though this should be interpreted cautiously. During experiments, some participants quickly adapted, allowing the robot to assist and thereby maximizing efficiency. Others, however, resisted the robot's guidance, often overcompensating and increasing task duration, even when the robot was aligned with their intent. Notably, the number of failures did not significantly decrease with shared autonomy. This may be due to the absence of an autonomous grasping mechanism capable of completing the task when the robot is close to the object and goal confidence is high. Failures often stemmed from users misjudging distances or failing to correctly align grasps. In this paper, we assumed that each landmark was correctly labeled to its corresponding goal. However, incorrect labeling could lead to corrupted predictions. Fortunately, since goals are typically associated with multiple landmarks, the impact of a single mislabeled landmark is often attenuated. Additionally, the introduction of the `Undecided` state in the HMM helps mitigate the consequences of incorrect goal predictions: the probability of entering this state increases when there is a conflict between the user's intended goal and the one predicted, particularly when the discrepancy arises from specific user motions. A key limitation of our approach, as with most shared autonomy methods, is the reliance on a predefined, hard-coded goal set for the recognition algorithm. Large Language Models (LLMs) offer a promising direction here, as they can identify landmarks for a wide range of goals, especially in high-level planning contexts [23]. We are actively exploring this with the aid of object recognition tools like YOLO, and early results are encouraging. Visual Language Action (VLA) models such as Open-VLA [24] further demonstrate that real-time execution is feasible.

TABLE II

AVERAGE NUMBER OF TIMES PARTICIPANTS CAUSED AN OBJECT TO FALL FROM THE TABLE (FAILURES) AND MANIPULATION TIME FOR PARTICIPANTS USING THE SHARED AUTONOMY CONTROLLER COMPARED TO THE BASIC CONTROLLER, INCLUDING STANDARD DEVIATION ACROSS THREE TRIALS WITH SHARED AUTONOMY AND THREE WITHOUT.

Participant	1	2	3	4	5	6	7	8	9	10	All
Failures Basic	2	3.67	3	4.67	2.67	1	2.33	2	2	2.33	2.57
Average Basic (s)	329.69	301.68	365.59	521.68	277.82	259.19	257.45	285.97	283.38	303.30	318.58
SD Basic (s)	61.14	99.39	105.89	31.01	45.65	41.33	76.2	29.34	34.23	62.24	92.58
Failures Shared autonomy	4	3.67	3.33	4	2.33	1	1.67	1.33	2	1.67	2.5
Average Shared autonomy (s)	251.53	196.05	267.80	348.64	210.79	182.29	193.10	222.93	213.13	213.46	229.97
SD Shared autonomy (s)	35.41	28.27	120.3	55.96	88.43	44.15	16.17	20.98	54.19	31.25	67.79

ACKNOWLEDGMENT

This paper is based on results obtained from a project of Programs for Bridging the gap between R&D and the IDeal society (society 5.0) and Generating Economic and social value (BRIDGE)/Practical Global Research in the AI × Robotics Services, implemented by the Cabinet Office, Government of Japan, and partially funded by the Japan Science and Technology Agency (JST) with the JST-Mirai Program, grant number JPMJMI21H4.

The absence of an ethics committee for this experiment is justified primarily because it involves participants who are colleagues within the same institution, engaged in a collaborative internal project. This study, therefore, falls under the institution's internal guidelines and procedures, which provide an alternative framework for ensuring ethical standards. As the project does not involve external participants or require the handling of sensitive information, it aligns with standard practices for internal research and development activities.

REFERENCES

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

[2] S. Jain and B. Argall, "Probabilistic human intent recognition for shared autonomy in assistive robotics," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 9, no. 1, pp. 1–23, 2019.

[3] Z. Li, S. Zhao, J. Duan, C.-Y. Su, C. Yang, and X. Zhao, "Human cooperative wheelchair with brain–machine interaction based on shared control strategy," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 185–195, 2016.

[4] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," *Robotics science and systems: online proceedings*, vol. 2015, 2015.

[5] J. Hong, "Goal recognition through goal graph analysis," *Journal of Artificial Intelligence Research*, vol. 15, pp. 1–30, 2001.

[6] M. Ramırez and H. Geffner, "Goal recognition over pomdps: Inferring the intention of a pomdp agent," in *IJCAI*. Citeseer, 2011, pp. 2009–2014.

[7] Y. E-Martín, M. D. R-Moreno, and D. E. Smith, "A fast goal recognition technique based on interaction estimates," in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 761–768.

[8] M. Vered and G. A. Kaminka, "Heuristic online goal recognition in continuous domains," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 4447–4454.

[9] R. Pereira, N. Oren, and F. Meneguzzi, "Landmark-based heuristics for goal recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[10] R. F. Pereira, N. Oren, and F. Meneguzzi, "Landmark-based approaches for goal recognition as planning," *Artificial Intelligence*, vol. 279, p. 103217, 2020.

[11] A. D. Dragan and S. S. Srinivasa, *Formalizing assistive teleoperation*. MIT Press, July, 2012, vol. 376.

[12] ——, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[13] 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.

[14] R. M. Aronson, T. Santini, T. C. Kübler, E. Kasneci, S. Srinivasa, and H. Admoni, "Eye-hand behavior in human-robot shared manipulation," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 4–13.

[15] G. Synnaeve and P. Bessiere, "A bayesian model for plan recognition in rts games applied to starcraft," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2011.

[16] C. L. Baker and J. B. Tenenbaum, "Modeling human plan recognition using bayesian theory of mind," *Plan, activity, and intent recognition: Theory and practice*, vol. 7, pp. 177–204, 2014.

[17] F. R. Meneguzzi and R. F. Pereira, "A survey on goal recognition as planning," in *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), 2021, Canadá.*, 2021.

[18] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[19] R. Cisneros-Limón, A. Dallard, M. Benallegue, K. Kaneko, H. Kaminaga, P. Gergondet, A. Tanguy, R. P. Singh, L. Sun, Y. Chen *et al.*, "A cybernetic avatar system to embody human telepresence for connectivity, exploration, and skill transfer," *International Journal of Social Robotics*, pp. 1–28, 2024.

[20] R. P. Singh, P. Gergondet, and F. Kanehiro, "mc-mujoco: Simulating articulated robots with fsm controllers in mujoco," in *2023 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2023, pp. 1–5.

[21] A. Belardinelli, "Gaze-based intention estimation: principles, methodologies, and applications in hri," *ACM Transactions on Human-Robot Interaction*, vol. 13, no. 3, pp. 1–30, 2024.

[22] T. B. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Transactions on robotics and Automation*, vol. 9, no. 5, pp. 592–606, 1993.

[23] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2998–3009.

[24] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.