Shared Autonomy with Learned Latent Actions

Hong Jun Jeon
Stanford University
hjjeon@stanford.edu

Dylan P. Losey
Stanford University
dlosey@stanford.edu

Dorsa Sadigh
Stanford University
dorsa@cs.stanford.edu

Abstract—**Assistive robots enable people with disabilities to conduct everyday tasks on their own. However, these tasks can be complex, containing both coarse reaching motions and fine-grained manipulation. For example, when eating, not only does one need to move to the correct food item, but they must also precisely manipulate the food in different ways (e.g., cutting, stabbing, scooping). Shared autonomy methods make robot teleoperation safer and more precise by arbitrating user inputs with robot controls. However, these works have focused mainly on the high-level task of *reaching* a goal from a discrete set, while largely ignoring *manipulation* of objects at that goal. Meanwhile, dimensionality reduction techniques for teleoperation map useful high-dimensional robot actions into an intuitive low-dimensional controller, but it is unclear if these methods can achieve the requisite *precision* for tasks like eating. Our insight is that—by combining intuitive embeddings from learned latent actions with robotic assistance from shared autonomy—we can enable *precise assistive manipulation*. In this work, we adopt learned latent actions for shared autonomy by proposing a new model structure that changes the meaning of the human's input based on the robot's confidence of the goal. We show convergence bounds on the robot's distance to the most likely goal, and develop a training procedure to learn a controller that is able to move between goals even in the presence of shared autonomy. We evaluate our method in simulations and an eating user study. See videos of our experiments here: https://youtu.be/7BouKojzVyk.**
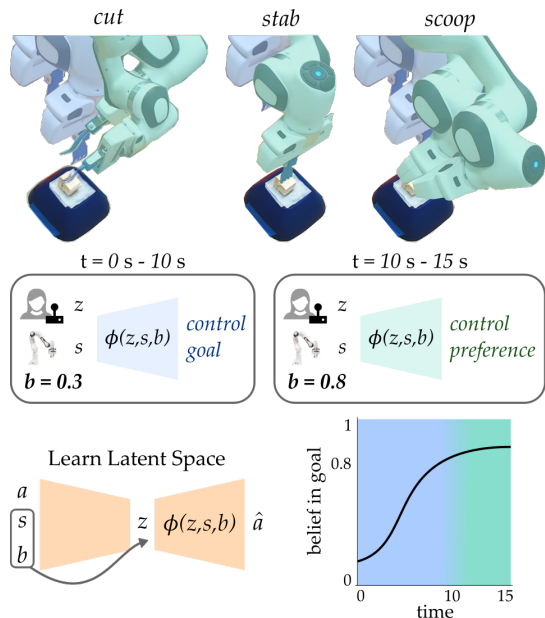
Fig. 1. User teleoperating an assistive robot to perform eating tasks. The human starts by controlling the robot's high-level motion. As the robot gets more confident about the human's goal, the meaning of the inputs becomes more refined, and the human precisely adjusts the robot's movement.

I. INTRODUCTION

There are nearly one million American adults living with physical disabilities that need external assistance when eating [32]. Physically assistive robots—such as wheelchair-mounted robotic arms—promise to help these people eat independently, without relying on caregivers [14, 21]. We envision a future where users teleoperate assistive robots (e.g., through joysticks [12], sip-and-puffs [1], or brain-computer interfaces [22]) to seamlessly perform complex and dexterous eating tasks.

For instance, imagine that you are controlling an assistive robotic arm to get a bite of tofu. You have a high-level goal: you want to guide the robot to reach for the tofu on the table in front of you. But just reaching the tofu is not enough; once there, you also need to *precisely* manipulate the arm to cut off a piece and pick it up with your fork (see Fig. 1). An effective robotic partner should assist with both the high-level reaching motions and the fine-grained manipulation tasks.

We will refer to the human's high-level objectives as *goals* (e.g., reaching the tofu), and their low-level manipulation as *preferences* (e.g., cutting, stabbing, or scooping). Completing a task according to your goals and preferences is challenging—particularly because today's assistive robots are teleoperated

using *low-dimensional* interfaces [12, 1, 22], while these tasks require high-dimensional, coordinated, and precise control.

Shared autonomy can make eating tasks easier by predicting the human's intent and then augmenting their input [16, 8, 15]. But current works focus on *goals*—i.e., helping the human move the robot between a set of discrete options—and do not provide assistance *after* reaching the goal, when the human must control the robot along a continuum of preferences.

Another approach is to develop better interfaces for directly controlling the robot. For example, the robot can use demonstrations to *learn* an intelligent mapping between the human's low-dimensional inputs and the robot's high-dimensional actions [19]. This approach makes sense when moving along the continuum of *preferences*; but the robot does not provide any additional guidance, so that any *imperfect or noisy* human inputs will move the robot away from the precise goal.

Viewed together, humans must be able to intuitively control the robot while performing complex and precise manipulation tasks. These tasks involve moving between discrete, high-level goals and fine-tuning along continuous, low-level preferences.

We combine learning intuitive embeddings with shared autonomy to enable precise assistive manipulation.

We view these methods as complementary types of assistance. Shared autonomy *constrains* the robot to high-level goals, while intuitive mappings *embed* the robot's motion into a sub-manifold of preferences. Returning to our eating example: the human starts by guiding the robot arm towards the tofu. As the robot becomes more confident about this goal, shared autonomy completes the motion, and the user's inputs transition to control the robot's fine-grained preferences (see. Fig. 1).

Overall, we make the following contributions:

Controlling Goals and Preferences. We formalize assistive tasks with goals and preferences and introduce the properties that intuitive interfaces should have during these tasks. Next, we propose a new model structure where the meaning of the human's inputs changes based on the robot's confidence.

Balancing Convergence with Change. We theoretically identify a convergence bound on the robot's distance from the most likely goal. To ensure that users are not trapped at the wrong goal, we add a novel entropy term that encourages versatility, and test this with a spectrum of simulated humans.

Conducting User Study with Eating Task. Participants teleoperate a 7 degree-of-freedom (*DoF*) robot arm with a 2-DoF joystick. We compare our approach to state-of-the-art baselines, and measure precision with and without both shared autonomy and the intuitive embedding.

II. RELATED WORK

We combine latent actions—a representation learning technique for discovering low dimensional input spaces for controlling a high dimensional system—with shared autonomy, a well established paradigm for incorporating both human and robot inputs to control a system.

Application Area – Assistive Robotics. Among the challenges of eating is the ability to prepare and transport food [14], a skill that assistive robotics has tried to enable. Researchers have developed robot policies that can autonomously manipulate and deliver different types of food to users with disabilities [10, 25]. However, designing a fully autonomous system to handle a task as variable and personalized as eating is exceedingly challenging: one size *does not* fit all. We therefore develop a *partially* autonomous algorithm that allows users to perform precise manipulation tasks with a robotic arm.

Shared Autonomy. Our partially autonomous system falls under shared autonomy, a framework in which the robot receives human user inputs and combines them appropriate autonomous inputs for a safer and more efficient outcome. Many shared autonomy algorithms provide robotic assistance to users who must reach for a goal objects in the environment [8, 16, 22, 11, 2, 13]. Several of these works infer human intent by maintaining a belief over the set of possible goals, and using human inputs as evidence in a Bayesian framework to continually update this belief [8, 16, 22, 29, 30]. The algorithms often apply more assistance as confidence in a goal increases.

Other works in shared autonomy propose or learn suitable dynamics models to translate user inputs to robot actions [26,

28, 27]. The learned latent actions in our approach mirror the work of Reddy *et. al* [28], but have the added challenge of mapping a *low* dimensional input to a *high* dimensional action.

Latent Actions. Controlling robots is difficult when they contains many degrees of freedom (DoFs). To control the entire system, joysticks often include a button to toggle between different modes, each of which controls a subset of the full system. This can be extremely taxing for the users [12]. Prior work has tried to prune away unnecessary control axes through Principal Components Analysis (PCA) [5, 3]. More recently, researchers have learned *non-linear* mappings to control high dimensional robot arms through low dimensional inputs [19]. There is also extensive work in learning latent representations for components in RL [4, 6, 33, 20, 9]. These methods use autoencoders [18, 7] to learn mappings between structurally sparse high dimensional data and low dimensional embedding spaces without supervision. We build on work in learned latent actions [19] and task-conditioning [24] to learn an embedding for teleoperation *in the presence* of shared autonomy.

III. USING LATENT ACTIONS & SHARED AUTONOMY

We explore tasks where the robot's movements naturally become more refined and precise over time. Recall our eating example: users start by indicating where the robot should go (e.g., *reach for the tofu*), and then control what the robot does at that goal (e.g., *cut off a piece and pick it up with the fork*). Completing these tasks is particularly challenging with assistive robots, where the user must interact with a low-dimensional control interface (e.g., a joystick).

Overview. In this section, we propose an algorithm that *refines* the robot's assistance. At the start of the task, the human's low-DoF inputs coarsely move the robot towards a high-level *goal* (e.g., reaching the tofu). Once the robot reaches this goal, the low-DoF inputs change meaning to precisely manipulate along the human's low-level *preferences* (e.g., cutting a piece). We leverage latent actions to learn this changing mapping: specifically, we learn a decoder $\phi(\cdot)$ that enables the human to control a spectrum of goal-directed motions and fine-grained preferences. In order to guide the robot to the user's goal—and maintain this goal while the human focuses on conveying their preference—we apply shared autonomy.

Formulation. We formulate the human's task as a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{G}, \Theta, R, \gamma \rangle$. Let $s \in \mathcal{S} \subseteq \mathbb{R}^n$ be the robot's state and let $a \in \mathcal{A} \subseteq \mathbb{R}^m$ be the robot's action: when the robot takes action a in state s, it transitions according to $\mathcal{T}(s, a)$. The human has a high-level goal $g^* \in \mathcal{G}$ and low-level preference $\theta^* \in \Theta$. Together, the goal (e.g., *reaching the tofu*) and preference (e.g., *cutting the tofu*) determine the robot's reward function: the robot should maximize $R(s, g^*, \theta^*)$ with the discount factor $\gamma \in [0, 1)$.

The space of candidate goals $\mathcal{G}$ is discrete and *known* by the robot *a priori*. We let $b \in \mathcal{B}$ denote the robot's belief over this space of candidate goals, where $b(g) = 1$ indicates that the robot is convinced that g is the human's desired goal.

The space of preferences Θ is continuous and *unknown* by the robot. We do not maintain a belief here; instead, we assume

the robot has access to $\mathcal{D}$, a dataset of relevant demonstrations (e.g., examples of reaching for and then manipulating the tofu). These demonstrations consist of state-action-belief tuples: $\mathcal{D} = \{(s^0, a^0, b^0), (s^1, a^1, b^1), \ldots\}$.

Dynamics. The human teleoperates the robot by using a low-dimensional interface (e.g., a joystick). Let $z \in \mathcal{Z} \subset \mathbb{R}^d$ be the human's input—where $d < m$—and let $\phi(\cdot)$ be a *decoder* function that maps these low-dimensional human inputs into the robot's action space, $\mathcal{A}$. We denote the resulting action as $a_h \in \mathcal{A}$. The robot's overall behavior combines this input action and $a_r \in \mathcal{A}$, the robot's assistive guidance [8, 23]:

$$a = (1 - \alpha) \cdot a_h + \alpha \cdot a_r \tag{1}$$

$\alpha \in [0, 1]$ parameterizes the trade-off between direct teleoperation ($\alpha = 0$) and complete autonomy ($\alpha = 1$).

Problem Statement. Our objective is to intuitively *decode* the human's low-DoF inputs by learning $\phi(\cdot)$, and then combine these inputs with the robot's *autonomous* high-DoF actions a_r. This problem is made challenging by the need to assist users as they control dexterous robots along coarse, goal-directed movement and precise, preferred manipulations.

A. Learned Latent Actions (LA)

Latent actions refer to low-dimensional representations of high-dimensional actions that are learned through dimensionality reduction techniques [19, 20, 17]. Given a set of demonstrated motions, the robot embeds the high-DoF actions into a latent action space, and then decodes these latent actions to reconstruct the original action (see Fig. 1, bottom). Previous works have leveraged latent actions for intuitive low-DoF control of assistive robots, where latent actions enable users to express their desired high-DoF motion [19, 5, 3].

Unlike these prior works, we recognize that often the *meaning* of latent actions changes within a precision task. Imagine that you are using a 1-DoF interface to perform the task in our eating example. At the start of the task, you need pressing left and right on the joystick (i.e., z) to move the robot towards the tofu. But as the robot approaches the tofu, you no longer need to keep moving towards a goal; instead, you need to use those same joystick inputs to carefully align the orientation of the fork, so that you can cut off a piece.

While latent actions provide an intuitive way to convey the user's intent, there are often more actions to convey than the latent space can embed. We thus need latent actions that can *convey different types of meanings*—indicating the human's goal, preference, or some combination of the two.

Conditioning on Belief. In order to learn latent action spaces that can continuously alternate between controlling high-level goals and fine-grained preferences, we will condition on the robot's current *context*. This context includes the state s – the configuration that the robot is in – as well as the *belief* b – the robot's confidence in the goal. Intuitively, conditioning on belief enables the *meaning* of latent actions to change based on the robot's confidence. As a result of this proposed structure, latent actions *purely indicate the desired goal* when the robot is

unsure; and once the robot is confident about the human's goal, latent actions change to *convey the preferred manipulation*. We design and enforce models that decode the meaning of latent actions based on context: $\phi : \mathcal{Z} \times \mathcal{S} \times \mathcal{B} \to \mathcal{A}$.

Reconstructing Actions. We now have a robot that decodes the user's input based on its state and confidence; but how do we ensure the decoded actions are *accurate*? Put another way, how do we ensure that the robot learns latent actions that can actually move the high-dimensional robot towards the human's goal and then correctly manipulate the object? To resolve this problem, we turn to the dataset $\mathcal{D}$, which contains examples of these desired, high-DoF actions. Specifically, consider the state-action-belief tuple $(s, a, b) \in \mathcal{D}$: we want to learn a latent action space $\mathcal{Z}$ such that given s, b, and some $z \in \mathcal{Z}$, the robot reconstructs the demonstrated action a. Let $a_h \in \mathcal{A}$ be the reconstructed action, where $a_h = \phi(z, s, b)$, and let $e_a = a_h - a$ be the error between the reconstructed and demonstrated actions. To learn latent spaces that accurately reconstruct the demonstrated actions, we explore models that actively attempt to minimize the reconstruction error $\|e_a\|^2$ in the loss function.

B. Latent Actions with Shared Autonomy (LA + SA)

Latent actions provide an expressive mapping between low-dimensional user inputs and high dimensional robot actions. But controlling a robot with latent actions alone still presents a challenge: any *imprecision* or *noise* in either the user's inputs or latent space is reflected in the decoded actions. Recall our eating example: when the user is trying to guide the robot towards their preferred cutting motion, their inputs should not unintentionally cause the robot arm to drift away from the tofu or suddenly jerk into the table. Accordingly, we here leverage shared autonomy to facilitate *precise* robot motions, which assist the human towards their goals, and then maintain these goals as the human focuses on their preferences.

Providing Assistance. Recall that the robot applies assistance via action a_r in Eq. (1). In order to assist the human, the robot needs to understand the human's *intent*—i.e., which goal they want to reach. The robot's understanding of the human's intended goal is captured by the belief b, and we can leverage this belief to select an assistive action a_r. Similar to [8] and [16], let the robot provide assistance towards each discrete goal $g \in \mathcal{G}$ in proportion to the robot's confidence in that goal[1]:

$$a_r = \sum_{g \in \mathcal{G}} b(g) \cdot (g - s) \tag{2}$$

So now—if the robot has a uniform prior over which morsel of food the human wants to eat—a_r guides the robot to the center of these morsels. And—when the human indicates a desired morsel—a_r guides the robot towards that target.

Algorithm. Our approach for combining latent actions (*LA*) with shared autonomy (*SA*) is summarized in Algorithm 1. We emphasize that *LA+SA* is different from either latent actions or shared autonomy alone. Without latent actions, a robot using

---

[1]Our approach is not tied on this particular instantiation of shared autonomy. Other instances of shared autonomy can similarly be used.

Algorithm 1 Latent Actions & Shared Autonomy (*LA+SA*)

1: Given a discrete set of goals $\mathcal{G}$ and a dataset of example trajectories $\mathcal{D} = \{(s^0, a^0, b^0), (s^1, a^1, b^1), \ldots\}$
2: Train a model on $\mathcal{D}$ to learn the decoder $\phi(z, s, b)$
3: **for** $t \leftarrow 1, 2, \ldots, T$ **do**
4:     Set z^t as the human's low-DoF input
5:     $a_h^t \leftarrow \phi(z^t, s^t, b^t)$ ▷ map z^t to high-DoF action
6:     $a_r^t \leftarrow \sum_{g \in \mathcal{G}} b^t(g) \cdot (g - s^t)$ ▷ get robot assistance
7:     $a^t \leftarrow (1 - \alpha) \cdot a_h^t + \alpha \cdot a_r^t$ ▷ blend both a_h and a_r
8:     $b^{t+1} \propto P(a_h | s, g) P(g)$ ▷ update belief over goals
9:     $s^{t+1} \sim \mathcal{T}(s^t, a^t)$ ▷ take action and transition states
10: **end for**

shared autonomy must rely on predetermined, one-size-fits-all mappings from z to a_h. Without shared autonomy, latent actions require perfect teleoperation to reach and maintain goals. Put another way: shared autonomy constrains the robot to goals, while latent actions embed the robot's motions into a continuous sub-manifold of preferences.

IV. THEORETICAL ANALYSIS

We propose *LA+SA* as an approach for tasks involving goals and preferences. Both latent actions and shared autonomy have an *independent* role within this method: but how can we be sure that the *combination* of these tools will remain effective? Returning to our eating example—if the human inputs latent actions, will shared autonomy correctly guide the robot to the desired morsel of food? What if the human has multiple goals in mind (e.g., getting a chip and then dipping it in salsa)—can the latent actions change goals even when shared autonomy is confident? And what if the environment changes—how do we transfer the learned latent actions to new goals?

A. Converging to the Desired Goal

We first explore how *LA+SA* ensures that the human reaches their desired goal. Consider the Lyapunov function:

$$V(t) = \frac{1}{2}\|e(t)\|^2, \quad e(t) = g^* - s(t) \tag{3}$$

where e denotes the error between the robot's current state s and the human's goal g^*. We want the robot to choose actions that minimize Eq. (3) across a spectrum of user skill levels and teleoperation strategies. Let us focus on the common setting in which s is the robot's joint position and a is the joint velocity, so that $\dot{s}(t) = a(t)$. Taking the derivative of Eq. (3) and substituting in this transition function, we reach[2]:

$$\dot{V}(t) = -\frac{1}{2}e^\top \left[\phi(z, s, b) + \sum_{g \in \mathcal{G}} b(g) \cdot (g - s)\right] \tag{4}$$

We want Eq. (4) to be negative, so that V (and thus the error e) decrease over time. A sufficient condition for $\dot{V} < 0$ is:

$$b(g^*) \cdot \|e\| > \|\phi(z, s, b)\| + \sum_{g \in \mathcal{G}'} b(g) \cdot \|g - s\| \tag{5}$$

[2]For notational simplicity we choose $\alpha = 0.5$, so that both human and robot inputs are equally weighted. Our results generalize to other α.

where $\mathcal{G}'$ is the set of all goals except g^*. As a final step, we bound the magnitude of the decoded action, such that $\|\phi(\cdot)\| < \sigma_h$, and we define σ_r as the distance between s and the furthest goal: $\sigma_r = \max_{g \in \mathcal{G}'} \|g - s\|$. Now we have $\dot{V} < 0$ if:

$$b(g^*) \cdot \|e\| > \sigma_h + (1 - b(g^*)) \cdot \sigma_r \tag{6}$$

We define $\delta := \sigma_h + (1 - b(g^*)) \cdot \sigma_r$. We therefore conclude that *LA+SA* yields *uniformly ultimately bounded stability* about the human's goal, where δ affects the *radius* of this bound [31]. As the robot's confidence in g^* increases, $\delta \to \sigma_h$, and the robot's error e decreases so long as $\|e(t)\| > \sigma_h$. Intuitively, *LA+SA* guarantees that the robot will move to some ball around the human's goal g^*, and the radius of that ball decreases as the robot becomes more confident.

B. Changing Goals

Our analysis in Sec. IV-A suggests that the robot becomes *constrained* to a region about the most likely goal. This works well when the human correctly conveys their intentions to the robot—but what if the human makes a mistake, or changes their mind? How do we ensure that the robot is not *trapped* at an undesired goal? Re-examining Eq. (6), it is key that—at every (s, b) pair—the human can convey sufficiently large actions $\|\phi(z, s, b)\|$ towards their preferred goal, ensuring that σ_h does not decrease to zero. Put another way, the human must be able to *increase* the radius of the bounding ball, reducing the constraint imposed by shared autonomy.

To encourage the robot to learn latent actions that increase this radius, we introduce an additional term into our model's loss function. We reward the robot for learning latent actions that have high *entropy* with respect to the goals; i.e., in a given context (s, b) there exist latent actions z that cause the robot to move towards *each* of the goals $g \in \mathcal{G}$. Define $p_{(s,b)}(g)$ as proportional to the total *score* η accumulated for goal g:

$$p_{(s,b)}(g) \propto \sum_{z \in \mathcal{Z}} \eta(g, s, b, z) \tag{7}$$

where the score function η indicates how well action z taken from state (s, b) conveys the intent of moving to goal g, and the distribution $p_{(s,b)}$ over $\mathcal{G}$ captures the proportion of latent actions z at state (s, b) that move the robot toward each goal. Intuitively, $p_{(s,b)}$ captures the comparative ease of moving toward each goal: when $p_{(s,b)}(g) \to 1$, the human can easily move towards goal g since *all* latent actions at (s, b) induce movement towards goal g and consequently, *no* latent actions guide the robot towards any other goals. We seek to *avoid* learning latent actions where $p_{(s,b)}(g) \to 1$, because in these scenarios the teleoperator *cannot* correct their mistakes or move towards a different goal! Recall from Sec. III-A that the model should minimize the reconstruction error, $e_a = a_h - a$. We now argue that the model should additionally maximize the Shannon entropy of p, so that the loss function becomes:

$$\mathcal{L} = \|e_a\|^2 + \lambda \cdot \sum_{g \in \mathcal{G}} p(g) \log p(g) \tag{8}$$

Here the hyperparameter $\lambda > 0$ determines the relative trade-

off between reconstruction error and latent action entropy. For clarity, we emphasize that this loss function $\mathcal{L}$ is leveraged offline, when training a model to learn the decoder $\phi(\cdot)$.

C. Introducing New Goals

We have covered how the robot can reach and change goals during the task—but what about situations where new goals are introduced *dynamically*? For instance, imagine that in our eating scenario a new plate of food is set in front of the user. The decoder $\phi(\cdot)$ has already been trained using the dataset $\mathcal{D}$, which does not include example trajectories reaching for this plate. Accordingly, the latent action space may not contain actions that move the robot towards this new plate, preventing the human from interacting with this new goal!

We resolve this issue by leveraging the goals that the robot has already seen, *without collecting* new demonstrations *or retraining* the latent space. Let g be the new goal, and define $h(s, b) \rightarrow (\hat{s}, \hat{b})$ as a function that maps the robot's current context (s, b) to an *equivalent* context with respect to the previously seen goal $\hat{g}$. As a simple example, h could project the robot's current state s to a straight-line path between the start and g, and output the equivalent state $\hat{s}$ along the straight-line path between the start and $\hat{g}$ (while assigning the same confidence to $\hat{g}$ as the robot currently has over g). Using h, the overall process is as follows: (a) Convert to an equivalent context $(\hat{s}, \hat{b})$ where training data exists. (b) Decode the user's latent input in this equivalent context to identify the high-DoF action $\hat{a}_h = \phi(z, \hat{s}, \hat{b})$. (c) Transform $\hat{a}_h$ back to the original state to obtain the commanded action a_h. Robots can harness the models they have already learned with newly added goals, so that—if the robot has learned to pour, scoop, and stir at one bowl—the human still has these same latent actions available at a second bowl that has just been introduced.

V. SIMULATIONS

Our theoretical analysis highlights the benefits of combining latent actions with shared autonomy. However, it is not clear how this approach will work when interacting with a *spectrum of different users*. In this section we test our algorithm in a controlled environment with simulated humans. We compare models for learning latent actions with and without shared autonomy, and we simulate teleoperators with various levels of expertise and learning rates. We will investigate if *different types of users* can interact with our algorithm to *precisely* reach and change goals consistent with their preferred trajectory.

Model. We test latent actions used by themselves (*LA*), as well as latent actions combined with shared autonomy (*LA+SA*). For both *LA* and *LA+SA* we learn the latent space with an autoencoder conditioned on state and belief (as described in Sec. III-A). Building on our analysis from Sec. IV-B, we also test *LA+SA+Entropy*, where the autoencoder leverages Eq. (8) to additionally reward entropy in the learned latent space.

Environments. We implement these models on both a *simulated* and a *real* robot. The simulated robot is a 5-DoF planar arm, and the real robot is a 7-DoF FrankaEmika. For both
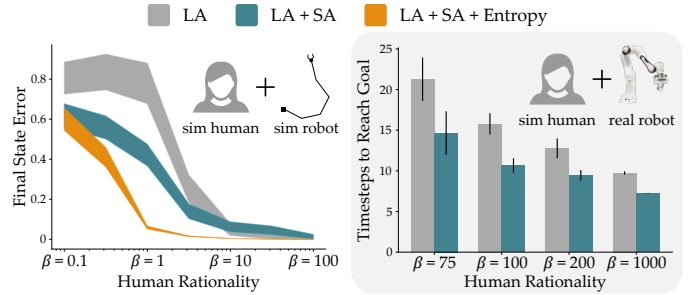

Fig. 2. Simulated humans for different levels of rationality. As $\beta \rightarrow \infty$, the human's choices approach optimal inputs. *Final State Error* (in all plots) is normalized by the distance between goals. Introducing shared autonomy (*SA*) improves the convergence of latent actions (*LA*), particularly when the human teleoperator is noisy and imperfect.

robots, the state s captures the current joint position, and the action a is a change in joint position, so that: $s^{t+1} = s^t + \Delta \cdot a^t$.

Task. We consider a manipulation task where there are two coffee cups in front of a robot arm. The human may want to reach either cup (i.e., goals), and grasp that cup along a continuous spectrum from the top to the side (i.e., preferences). We embed the robot's high-DoF actions into a 1-DoF input space: the simulated users had to convey both their goal and preference *only by pressing left and right* on the joystick.

Simulated Humans. The users attempting to complete this task are *approximately optimal*, and make decisions that guide the robot accordingly to their goal g^* and preference θ^*. Let s^* be the final pose that the human wants the robot to reach: s^* is based on both the position of their desired coffee cup (g^*) and the orientation of their preferred grasp (θ^*). The humans have reward function $R = -\|s^* - s\|^2$, and choose latent actions z to move the robot directly towards s^*:

$$p(z) \propto \exp\left\{ -\beta(t) \cdot \|s^* - (s + \phi(z, s, b))\|^2 \right\} \quad (9)$$

Within Eq. (9), $\beta \geq 0$ is a temperature constant that affects the user's *rationality*. When $\beta \rightarrow 0$, humans select increasingly random z, and when $\beta \rightarrow \infty$, humans always choose the z that moves the robot arm along their goal and preference. We simulate different types of users by varying $\beta(t)$.

A. Users with Fixed Expertise

We first simulate humans that have *fixed* levels of expertise. Here expertise is captured by β from Eq. (9): users with high β are proficient, and rarely make mistakes with noisy inputs. We anticipate that all algorithms will perform similarly when humans are always perfect or completely random—but we are particularly interested in the spectrum of users *between* these extremes, who frequently *mis-control* the robot.

Our results relating β to performance are shown in Fig. 2. In accordance with our convergence result from Sec. IV-A, we find that introducing shared autonomy helps humans reach their desired grasp more quickly, and with less final state error. The performance difference between *LA* and *LA+SA* decreases as the human's expertise increases—looking specifically at the real robot simulations, *LA* takes 45% more time to complete the task than *LA+SA* at $\beta = 75$, but only 30% more time
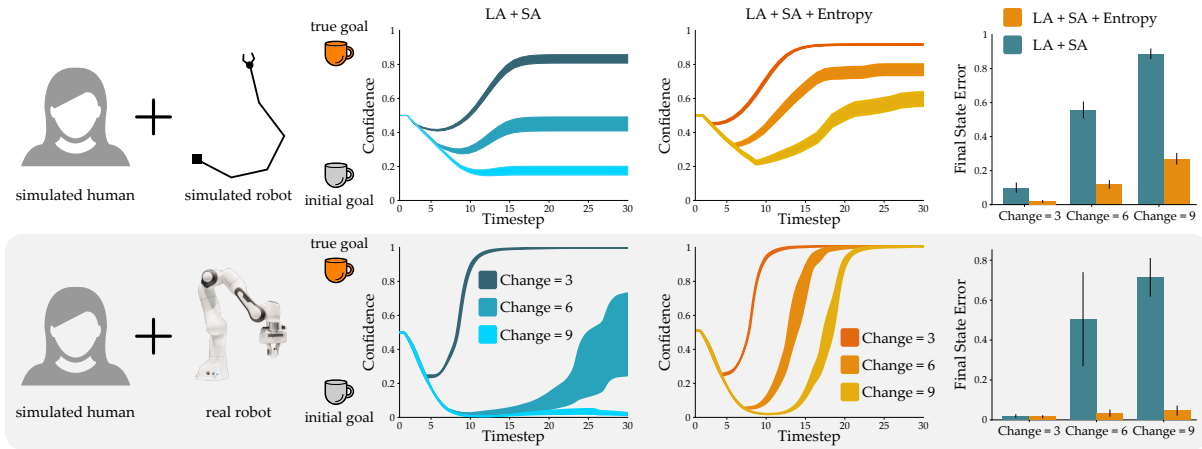
Fig. 3. Simulated humans that change their intended goal part-way through the task. *Change* is the timestep where this change occurs, and *Confidence* refers to the robot's belief in the human's true goal. Because of the constraints imposed by shared autonomy, users need latent actions that can overcome misguided assistance and move towards a less likely (but correct) goal. Encouraging entropy in the learned latent space (*LA+SA+Entropy*) enables users to switch goals.

when $\beta = 1000$. We conclude that shared autonomy improves performance across all levels of expertise, both when latent actions are trained with and without *Entropy*.

B. Users that Change their Mind

One downside of shared autonomy is over-assistance: like we discussed in Sec. IV-B, the robot may become *constrained* at likely (but incorrect) goals. To examine this adverse scenario we simulate humans that *change* which coffee cup they want to grasp after N timesteps. These simulated users intentionally move towards the *wrong* cup while $t \leq N$, and then try to reach the correct cup for the rest of the task. We model humans as near-optimal immediately after changing their mind about the goal, following Eq. (9) with a high β value.

We visualize our results in Fig. 3. When the latent action space is trained only to minimize reconstruction loss (*LA+SA*), users cannot escape the shared autonomy constraint around the wrong goal as N increases. Intuitively, this occurs because the latent space controls the intended goal when the belief b is roughly uniform, and then switches to controlling the preferred trajectory once the robot is confident. So if users change their goal after first convincing the robot, the latent space no longer contains actions that move towards this correct goal! We find that our proposed entropy loss function addresses this shortcoming: *LA+SA+Entropy* users are able to input actions z that alter the robot's goal. Our results support Sec. IV-B, and suggest that encouraging entropy at training time improves the robustness of the latent space.

C. Users that Learn within the Task

We not only expect real users to change their mind when collaborating with the robot, but we also anticipate that these teleoperators will *learn and improve* as they gain experience during the task. For instance, the user might learn that holding left on the joystick causes the robot to grasp the cup from the side, while holding right guides the robot towards a top grasp. To simulate this in-task learning, we set $\beta(t) = m \cdot t$, where the slope m determines how quickly the user learns. All users start with random actions ($\beta = 0$), and either learn *quickly*

(high m) or *slowly* (low m). We point out that slow learners may effectively "change their mind" multiple times, since they are unsure of how to control the robot.

Our findings are plotted in Fig. 4. Interestingly, we see that—for both fast and slow learners—*LA+SA+Entropy* improves in-task performance. We attribute this improvement to the inherent *versatility* of latent spaces that maximize entropy: as humans gain expertise, they can use these latent actions to quickly undo their mistakes and correct the robot's behavior.

D. Users Reaching for New Goals

So far the simulated humans are grasping coffee cups that the robot observed at training time. Here we introduce *new* coffee cups, and ask the users to reach for these goals *without retraining* the latent space. If we make no changes to our approach, the latent actions can only ever reach the original cup—there are no demonstrations that grasp this new goal! We therefore geometrically map the robot's current context— which is outside of the decoder distribution—into a equivalent context defined relative to a known goal—where the robot can actually decode z. Our results on the real robot are summarized in Fig. 5. We observe a *linear* relationship between distance and error ($y \approx 0.11x$, $R^2 = 0.98$): when the new coffee cup moves farther from the training cup, the robot's grasp becomes less accurate. But using goals as references has *reduced this error*: without mapping the context to a known goal, the robot only ever reaches for the training goal ($y = x$).

VI. USER STUDY

Motivated by the application of assistive robotics, we designed a user study with *eating tasks*. Participants teleoperated a 7-DoF robotic arm with a 2-DoF joystick to perform precise manipulation: users controlled the robot towards a goal plate, and then carefully adjusted the robot's motion to cut, stab, and scoop different foods (see Fig. 1).

Experimental Setup. Each participant attempted to complete two dishes: an *Entree task* and a *Dessert task*. In *Entree*, users had to perform multiple precise motions at the same goal. Here participants (a) guided the robot towards a bowl with tofu, (b)
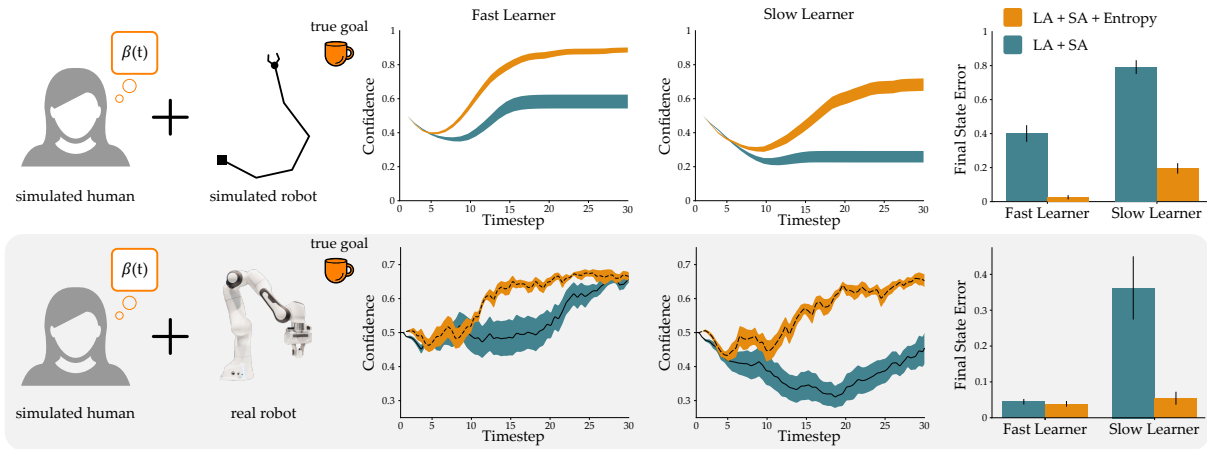
Fig. 4. Simulated humans that learn how to teleoperate the robot. The human's rationality $\beta(t)$ is linear in time, and either increases with a high slope (*Fast Learner*) or low slope (*Slow Learner*). As the human learns, they get better at choosing inputs that best guide the robot towards their true goal. We find that latent actions learned with the entropy reward (*LA+SA+Entropy*) are more versatile, so that the human can quickly undo mistakes made while learning.
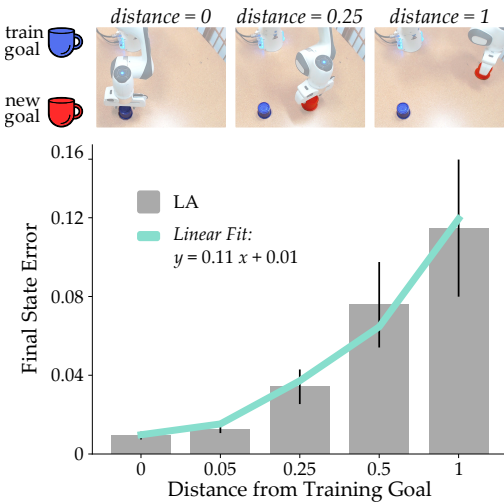

Fig. 5. Simulated humans teleoperating the real robot to reach new goals unseen at training time. Both axes are normalized by the distance between the training goal and the farthest new goal. The learned latent actions replicate demonstrations in the training dataset $\mathcal{D}$: we leverage known goals to *extend* these latent actions into contexts where no training data is provided.

cut off a slice of tofu, and (c) stabbed and scooped the slice onto their plate. In *Dessert* the participants had to convey their preferences at multiple goals: they (a) stabbed a marshmallow in the middle goal, (b) scooped it through icing at the right goal, and then (c) dipped it in rice at the left goal before (d) setting the marshmallow on their plate. In both tasks subjects sat next to the robot, mimicking a wheelchair-mounted arm.

Independent Variables. We conducted a 2x2 factorial design that separately varied *Control Interface* and *Robot Assistance*.

For the control interface, we tested a state-of-the-art direct teleoperation scheme (*Retargetting*), where the user's joystick inputs map to the 6-DoF end-effector twist of the robot [26]. We compared this direct teleoperation baseline to our learned *Latent Actions*: here the robot interprets the meaning of the human's inputs based on the current context.

For robot assistance, we tested *With* and *Without Shared Autonomy*. We implemented the shared autonomy algorithm

from [16], which assists the robot towards likely human goals. Crossing these factors, we totaled 4 different conditions:

- Retargeting (**R**)
- Retargetting + Shared Autonomy (**R+SA**)
- Latent Actions (**LA**)
- Latent Actions + Shared Autonomy (**LA+SA**)

The *LA+SA* condition is our proposed approach (Algorithm 1).

Model Training. We provided kinesthetic demonstrations $\mathcal{D}$ that guided the robot towards each plate, and then performed cutting, stabbing, and scooping motions at these goals. The robot learned the latent space (**LA**) from a total of 20 minutes of kinesthetic demonstrations.

Dependent Measures – Objective. We recorded the amount of time users took to complete each task (*Total Time*), as well as the amount of time spent without providing joystick inputs (*Idle Time*). We also computed proxy measures of the high-level goal accuracy and low-level preference precision. For goals, we measured the robot's total distance to the closest plate throughout the task (*Goal Error*). For preferences, we recorded the dot product between the robot's actual end-effector direction and the true end-effector directions needed to precisely cut, stab, and scoop (*Preference Alignment*).

Dependent Measures – Subjective. We administered a 7-point Likert scale survey after each condition. Questions were organized along five scales: how *Easy* it was to complete the tasks, how *Helpful* the robot was, how *Precise* their motions were, how *Intuitive* the robot was to control, and whether they would use this condition again (*Prefer*).

Participants and Procedure. We recruited 10 subjects from the Stanford University student body to participate in our study (4 female, average age 23.5 ± 2.15 years). All subjects provided informed written consent prior to the experiment. We used a within-subjects design: each participant completed both tasks with all four conditions (the order of the conditions was counterbalanced). Before every trial, users practiced teleoperating the robot with the current condition for up to 5 minutes.

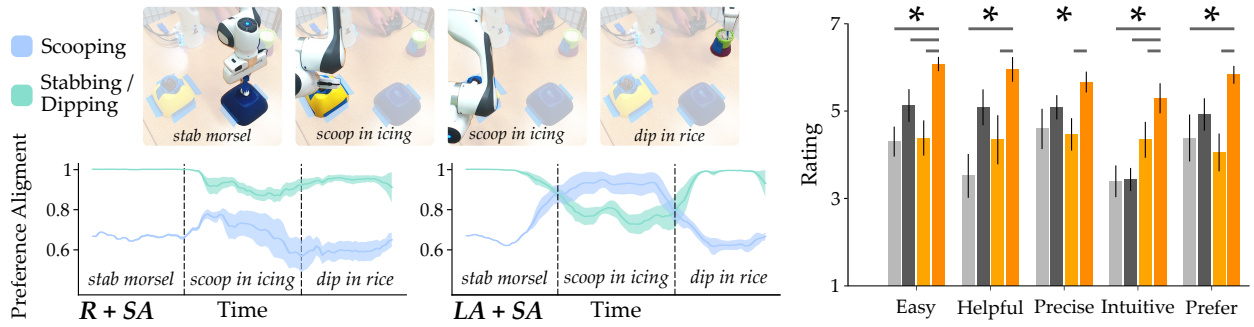**Hypotheses.** We tested three main hypotheses:

Fig. 6. (Left) The *Dessert* task consists of 3 phases: stabbing the morsel, scooping it in icing, and dipping it in rice. We identified the end-effector directions needed to complete these fine-grained preferences, and plotted the average dot product between the desired and actual directions (*Preference Alignment*). With **R+SA**, users executed the entire task in a stabbing/dipping orientation. By contrast, with **LA+SA**, users correctly adjusted the scooping preference in the second phase of the task. (Right) We plot the results of our 7-point Likert scale surveys. Color to method mappings are consistent with Fig. 7.

H1. *Users controlling the robot with Shared Autonomy (SA) will more accurately maintain their goals.*

H2. *Latent Actions (LA) will help users more precisely execute their preferences.*

H3. *Participants will complete the task most efficiently with combined LA+SA.*

Results – Objective. To explore **H1**, we analyzed the *Goal Error* for methods with and without **SA** (see Fig. 7). Across both tasks, users interacting with **SA** reached their intended goals *significantly more accurately* ($F(1, 18) = 29.9, p < .001$). Breaking this down by condition, users incurred *less* error with **LA+SA** than with **LA** ($p < .001$), and—similarly—users were more accurate with **R+SA** than with **R** ($p < .05$).

So shared autonomy helped users more accurately maintain their goals—but were participants able to complete the precise manipulation tasks at those goals? We visualize the *Preference Alignment* for *Dessert* in Fig. 6, specifically comparing **R+SA** to **LA+SA**. We notice that—when using direct teleoperation— participants remained in a stabbing preference throughout the task. By contrast, users with latent actions *adjusted* between preferences: stabbing the marshmallow, scooping it in icing, and dipping it in rice. These results support **H2**, suggesting that **LA** enables users to express their preferences.

Now that we know the benefits of **SA** and **LA** individually, what happens when we focus on their combination? Inspecting Fig. 8, participants using **LA+SA** were able to complete both tasks more efficiently. Summing times across both tasks, and then performing pair-wise comparisons between each condition, we found that **LA+SA** outperformed the alternatives for both *Total Time* ($p < .05$) and *Idle Time* ($p < .05$).

Results – Subjective. We find further support for **H3** in the user's feedback. The results of t-tests comparing **LA+SA** to the other conditions are reported in Fig. 6 (where an $*$ denotes $p < .05$). Responses suggest that users were most "comfortable" when performing precise manipulation with **LA+SA**.

Limitations. The 10 participants in our user study were *able-bodied*, and had equal practice time with each tested condition. By contrast, people applying these assistive control strategies on a daily basis will have significantly more experience and expertise, which could bias their skill towards conditions that
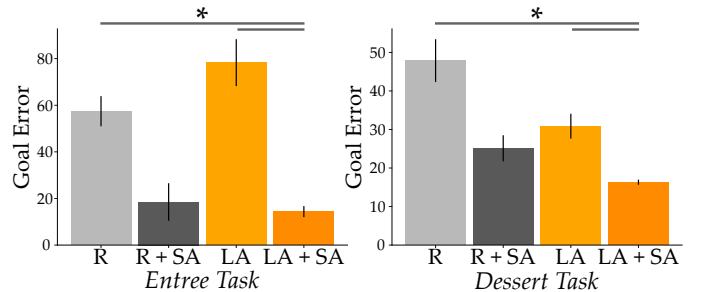

Fig. 7. Error between the nearest goal and the end-effector's current position. Adding **SA** decreased this error across both mapping strategies (**R** and **LA**).
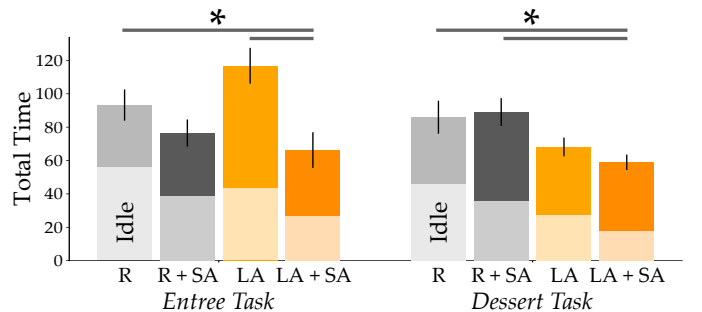

Fig. 8. Time taken to complete the task (solid) and time spent idle (light). Users completed both eating tasks most efficiently with **LA+SA**.

are more familiar. We also recognize that our target population will likely provide joystick inputs with higher noise levels.

VII. Conclusion

We focused on assistive teleoperation scenarios where the the human needs to control the robot's high-level goal and fine-grained motion. By combining shared autonomy with latent actions, we developed a method that constrains the human to their goal while embedding the robot's actions to precise sub-manifolds. Importantly, the meaning of the human's inputs changes as the robot becomes more confident—refining from coarse movements to careful adjustments. Our simulations and user studies suggest that this combined approach enables users to efficiently complete dexterous eating tasks.

References

[1] Brenna D Argall. Autonomy in rehabilitation robotics: An intersection. *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

[2] Reuben M. Aronson, Thiago Santini, Thomas C. Kübler, Enkelejda Kasneci, Siddhartha Srinivasa, and Henny Admoni. Eye-hand behavior in human-robot shared manipulation. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, page 4–13, 2018.

[3] P. K. Artemiadis and K. J. Kyriakopoulos. Emg-based control of a robot arm using low-dimensional embeddings. *IEEE Transactions on Robotics*, 26(2):393–398, 2010.

[4] Yash Chandak, Georgios Theocharous, James Kostas, Scott Jordan, and Philip Thomas. Learning action representations for reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 941–950, 2019.

[5] Matei T Ciocarlie and Peter K Allen. Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28(7):851–867, 2009.

[6] John D. Co-Reyes, Yuxuan Liu, Abhishek Gupta, Benjamin Eysenbach, Pieter Abbeel, and Sergey Levine. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. *CoRR*, 2018.

[7] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[8] Anca D Dragan and Siddhartha S Srinivasa. A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805, 2013.

[9] Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L. Isbell Jr. Imitating latent policies from observation. *CoRR*, 2018.

[10] Ryan Feng, Youngsun Kim, Gilwoo Lee, Ethan K Gordon, Matt Schmittle, Shivaum Kumar, Tapomayukh Bhattacharjee, and Siddhartha S Srinivasa. Robot-assisted feeding: Generalizing skewering strategies across food items on a realistic plate. In *International Symposium on Robotics Research*, 2019.

[11] Deepak Gopinath, Siddarth Jain, and Brenna D Argall. Human-in-the-loop optimization of shared autonomy in assistive robotics. *IEEE Robotics and Automation Letters*, 2(1):247–254, 2016.

[12] Laura V Herlant, Rachel M Holladay, and Siddhartha S Srinivasa. Assistive teleoperation of robot arms via automatic time-optimal mode switching. In *Int. Conf. on Human Robot Interaction (HRI)*, pages 35–42, 2016.

[13] Chien-Ming Huang and Bilge Mutlu. Anticipatory robot control for efficient human-robot collaboration. In *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*, pages 83–90. IEEE, 2016.

[14] Catrine Jacobsson, Karin Axelsson, Per Olov Österlind, and Astrid Norberg. How people with stroke and healthy older people experience the eating process. *Journal of clinical nursing*, 9(2):255–264, 2000.

[15] Siddarth Jain and Brenna Argall. Probabilistic human intent recognition for shared autonomy in assistive robotics. *ACM Transactions on Human-Robot Interaction (THRI)*, 9(1):1–23, 2019.

[16] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S Srinivasa, and J Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. *The International Journal of Robotics Research*, 37(7):717–742, 2018.

[17] Rico Jonschkowski and Oliver Brock. State representation learning in robotics: Using prior knowledge about physical interaction. In *Robotics: Science and Systems*, 2014.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Int. Conf. on Learning Representations (ICLR)*, 2014.

[19] Dylan P Losey, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, and Dorsa Sadigh. Controlling assistive robots with learned latent actions. *arXiv preprint arXiv:1909.09674*, 2019.

[20] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. *CoRR*, 2019.

[21] Tracy L Mitzner, Jon A Sanford, and Wendy A Rogers. Closing the capacity-ability gap: Using technology to support aging with disability. *Innovation in aging*, 2(1): igy008, 2018.

[22] Katharina Muelling, Arun Venkatraman, Jean-Sebastien Valois, John E Downey, Jeffrey Weiss, Shervin Javdani, Martial Hebert, Andrew B Schwartz, Jennifer L Collinger, and J Andrew Bagnell. Autonomy infused teleoperation with application to brain computer interface controlled manipulation. *Autonomous Robots*, 41(6): 1401–1422, 2017.

[23] Benjamin A Newman, Reuben M Aronson, Siddartha S Srinivasa, Kris Kitani, and Henny Admoni. Harmonic: A multimodal dataset of assistive human-robot collaboration. *arXiv preprint arXiv:1807.11154*, 2018.

[24] Michael Noseworthy, Rohan Paul, Subhro Roy, Daehyung Park, and Nicholas Roy. Task-conditioned variational autoencoders for learning movement primitives.

[25] Daehyung Park, Yuuna Hoshi, Harshal P Mahajan, Wendy A Rogers, and Charles C Kemp. Toward active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned. *arXiv preprint arXiv:1904.03568*, 2019.

[26] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. A motion retargeting method for effective mimicry-based teleoperation of robot arms. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 361–370, 2017.

[27] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. Where do you think you're going?: Inferring beliefs about dynamics from behavior. *CoRR*, 2018.

[28] Siddharth Reddy, Anca D Dragan, and Sergey Levine. Shared autonomy via deep reinforcement learning. In *RSS*, 2018.

[29] Dorsa Sadigh, S. Shankar Sastry, Sanjit A. Seshia, and Anca Dragan. Information gathering actions over human internal state. In *Proceedings of the IEEE, /RSJ, International Conference on Intelligent Robots and Systems (IROS)*, pages 66–73. IEEE, October 2016. doi: 10.1109/IROS.2016.7759036.

[30] Dorsa Sadigh, Nick Landolfi, S. Shankar Sastry, Sanjit A. Seshia, and Anca D. Dragan. Planning for cars that coordinate with people: Leveraging effects on human actions for planning and active information gathering over human internal state. *Autonomous Robots (AURO)*, 42(7):1405–1426, October 2018. ISSN 1573-7527. doi: 10.1007/s10514-018-9746-1.

[31] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*. 2006.

[32] Danielle M Taylor. *Americans With Disabilities*. US Census Bureau, 2018.

[33] Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin A. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *CoRR*, 2015.