Situational Confidence Assistance for Lifelong Shared Autonomy

Matthew Zurek*, Andreea Bobu*, Daniel S. Brown, and Anca D. Dragan

***Abstract*— Shared autonomy enables robots to infer user intent and assist in accomplishing it. But when the user wants to do a new task that the robot does not know about, shared autonomy will hinder their performance by attempting to assist them with something that is not their intent. Our key idea is that the robot can detect when its repertoire of intents is insufficient to explain the user's input, and give them back control. This then enables the robot to observe unhindered task execution, learn the new intent behind it, and add it to this repertoire. We demonstrate with both a case study and a user study that our proposed method maintains good performance when the human's intent is in the robot's repertoire, outperforms prior shared autonomy approaches when it isn't, and successfully learns new skills, enabling efficient lifelong learning for confidence-based shared autonomy.**

I. Introduction

In shared autonomy [1]–[11], robots assist human operators to perform their objectives more effectively. Here, rather than directly executing the human's control input, a typical framework has the robot estimate the human's intent and execute controls that help achieve it [2], [3], [12]–[14].

These methods succeed when the robot knows the set of possible human intents a priori, e.g. the objects the human might want to reach, or the buttons they might want to push [2], [12]. But realistically, users of these systems will inevitably want to perform tasks outside the repertoire of known intents – they might want to reach for a goal unknown to the robot, or perform a new task like pouring a cup of water into a sink. This presents a three-fold challenge for shared autonomy. First, the robot will be unable to recognize and help with something unknown. Second, and perhaps more importantly, it will attempt to assist with whatever wrong intent it infers, interfering with what the user is trying to do and hindering their performance. This happens when the robot plans in expectation [12], and, as our experiments will demonstrate, it happens even when the robot arbitrates the amount of assistance based on its confidence in the most likely goal [2]. Third, the new task remains just as difficult as the first time even after arbitrarily many attempts.

Our key idea is that the robot should detect that the user is trying something new and give them control. This then presents an opportunity for the robot to observe the new executed trajectory, learn the underlying intent that explains it, and add it to its repertoire so that it can infer and assist for this intent in the future.

To achieve this, we need two ingredients: 1) a way for the robot to detect its repertoire of intents is insufficient, and 2) a representation of intents that enables learning new tasks throughout its lifetime, adding them to its repertoire, and performing inference over them in a unified way with the initial known intents. For the latter, we use cost functions to unify goals and general skills like pouring into the same representation. This then enables the former: when the human acts too suboptimally for any of the known cost functions, it suggests the robot lacks the correct set of costs.

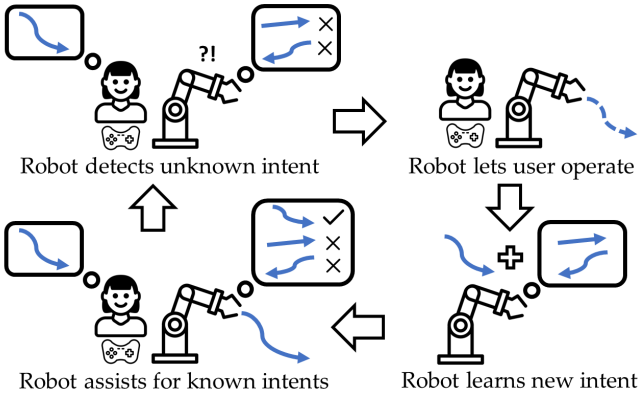

Fig. 1: We propose an approach for lifelong shared autonomy that enables a robot to detect when its set of known human intents is insufficient to explain the current human behavior. Rather than trying to assist for the wrong intent, the robot learns from novel teleoperations to learn a model of the new intent, allowing for lifelong confidence-based assistance.

Our approach takes inspiration from recent work on hypothesis misspecification where the robot recognizes when its cost function features are insufficient to explain human demonstrations and corrections [15], and updates the cost in proportion to the *situational confidence* in these features' ability to explain input. We extend detecting hypothesis mispecification to the context of shared autonomy, in which there are multiple intents, represented as cost functions, and the robot seeks to recognize whether any of the known intents explain the human input sufficiently. The robot can then arbitrate its assistance based on its confidence in the most likely intent being what the human wanted.

Our approach, which we call Confidence-Aware Shared Autonomy (CASA), allows the robot to ascertain whether the human inputs are associated with a known or new task. By arbitrating the user's input based on the confidence in the most likely intent, CASA follows a standard policy blending assistance approach if the task is known, and otherwise gives the user full control. Additionally, CASA allows the user to provide a few demonstrations of the new intent, which the robot uses to learn a cost function via Inverse Reinforcement

*Indicates equal contribution. Authors are with EECS at UC Berkeley. Research supported by the Air Force Office of Scientific Research (AFOSR), the Office of Naval Research (ONR), and NSF grant IIS1734633 (SCHooL).

Learning (IRL) [16] and add it to its set of intents. This enables lifelong shared autonomy, where the robot helps when it is confident in what the user wants and learns new intents when it detects that the human is doing something novel, so that it can assist with that intent in the future.

We test our approach in a expert case study and a user study with a simulated 7DoF JACO assistive robot arm. Our results suggest that CASA significantly outperforms prior approaches when assisting for unknown intents, maintains high performance in the case of known ones, and successfully learns new intents for better lifelong shared autonomy.

II. Confidence-Aware Shared Autonomy

We consider a human teleoperating a dexterous robotic manipulator to perform everyday manipulation tasks. The robot's goal is to assist the person in accomplishing their desired skill by augmenting or changing their input. While the robot possesses a predefined set of possible intents, the human's desired motion might not be captured by any of them. We propose that since the robot might not understand the person's intentions, it should reason about how confident it is in its predictions to avoid assisting for the wrong intent.

A. Preliminaries

Formally, let $x \in \mathcal{X}$ be the continuous robot state (e.g. joint angles), and $u \in \mathcal{U}$ the continuous robot action (e.g. joint velocity). The user controls their desired robot configuration by providing continuous inputs $a \in \mathcal{A}$ via an interface (e.g. GUI, joystick, keyboard commands, etc). These inputs are mapped to robot actions through a *direct teleoperation* function $\mathcal{T} : \mathcal{A} \to \mathcal{U}$. Define a person's trajectory up until time t as the sequence $\xi_{0\to t} = (x^0, a^0, \ldots, x^t, a^t)$.

The robot is equipped with a set of known intents Θ, one of which may represent the user's desired motion. Each intent is parameterized by a cost function C_θ, which may be hand-engineered or learned from demonstrations via IRL [17], [18]. For example, if the intent represents moving to a goal g, the cost function can be distance to the goal: $C_g(\xi) = \sum_{x\in\xi} \|x-g\|$. If the intent is pouring a cup, the cost can be a neural network with parameters ψ, C_ψ. Our shared autonomy system does not know the intent a priori, but infers it from the human's inputs. Given the user's trajectory so far, $\xi_{0\to t}$, a common strategy is to predict the user's intent $\theta \in \Theta$, compute the optimal action for moving accordingly, then augment the user's original input with it [2].

However, what if none of the intents match the human's input, i.e., the person is trying to do something the robot does not know about? We introduce a shared autonomy formalism where the robot reasons about its confidence in its current set of intents' ability to explain the person's input, and uses that confidence for robust assistance. This confidence serves a dual purpose, as the robot can also use it to ask the human to demonstrate the missing intent.

B. Intent Inference

To assist the person, the robot has to first predict which of its known tasks the person is trying to carry out, if any. To do that, the robot needs a model of how people teleoperate it to achieve a desired motion. We assume the Boltzmann noisily-rational decision model [19], [20]:

$$P(\xi \mid \theta, \beta) = \frac{e^{-\beta C_\theta(\xi)}}{\int_{\bar{\xi}} e^{-\beta C_\theta(\bar{\xi})} d\bar{\xi}} \ , \tag{1}$$

where the person chooses the trajectory ξ proportional to its exponentiated cost C_θ. The parameter $\beta \in [0, \infty)$ controls how much the robot expects to observe human input consistent with the intent θ. Typically, β is fixed, recovering the Maximum Entropy IRL observation model [17], which is what most inference-based shared autonomy methods use [2], [12]. Inspired by work on confidence-aware human-robot interaction [15], [21], [22], we instead reinterpret β as a measure of the robot's *situational confidence* in its ability to explain human data, given the known intents Θ, and we show how the robot can estimate it in Sec. II-C.

Given Eq. (1), if the cost C_θ of intent θ is additive along the trajectory ξ, we have that:

$$P(\xi_{0\to t} \mid \theta, \beta) = e^{-\beta C_\theta(\xi_{0\to t})} \frac{\int_{\bar{\xi}_{t\to T}} e^{-\beta C_\theta(\bar{\xi}_{t\to T})}}{\int_{\bar{\xi}_{0\to T}} e^{-\beta C_\theta(\bar{\xi}_{0\to T})}} \ , \tag{2}$$

where T is the duration of the episode. In high-dimensional manipulation spaces, evaluating these integrals is intractable. We follow [2] and approximate them via Laplace's method:

$$P(\xi_{0\to t} \mid \theta, \beta) \approx e^{-\beta(C_\theta(\xi_{0\to t}) + C_\theta(\xi^*_{t\to T}) - C_\theta(\xi^*_{0\to T}))} \times \sqrt{\left(\frac{\beta}{2\pi}\right)^{tk} \frac{|\nabla^2 C_\theta(\xi^*_{0\to T})|}{|\nabla^2 C_\theta(\xi^*_{t\to T})|}} \ , \tag{3}$$

where k is the action dimensionality, and the trajectories $\xi^*_{0\to T}$ and $\xi^*_{t\to T}$ are optimal with respect to C_θ and can be computed with any off-the-shelf trajectory optimizer[1].

Now, given a tractable way to compute the likelihood of the human input, the robot can obtain a posterior over intents:

$$P(\theta \mid \xi_{0\to t}, \beta) = \frac{P(\xi_{0\to t} \mid \theta, \beta)}{\sum_{\theta' \in \Theta} P(\xi_{0\to t} | \theta', \beta)}, \tag{4}$$

assuming $P(\theta \mid \beta) = P(\theta)$ and a uniform prior over intents.

Prior inference-based shared autonomy work [2], [12] typically assumes $\beta = 1$. We show that the robot should not be restricted by such an assumption and it, in fact, benefits from estimating $\hat{\beta}$ and reinterpreting it as a confidence.

C. Confidence Estimation

In the Boltzmann model in Eq. (1), we see that β determines the variance of the distribution over human trajectories. When β is high, the distribution is peaked around those trajectories ξ with the lowest cost C_θ; in contrast, a low β makes all trajectories equally likely. We can, thus, reinterpret β to take a useful meaning in shared autonomy: given an intent, β controls how well that intent's cost explains the user's input. A high β for an intent θ indicates that the intent's cost explains the input well and is a good candidate

[1]We use TrajOpt [23], based on sequential quadratic programming.

for assistance. A low β on all intents suggests that the robot's intent set is insufficient for explaining the person's trajectory.

We can thus estimate β and use it for assistance. Using the likelihood function in Eq. (3), we write the β posterior

$$P(\beta \mid \xi_{0\to t}, \theta) = \frac{P(\xi_{0\to t} \mid \theta, \beta)P(\beta)}{\int_{\bar{\beta}} P(\xi_{0\to t}|\theta, \bar{\beta})P(\beta)d\bar{\beta}}. \quad (5)$$

If we assume a uniform prior $P(\beta)$, we may compute an estimate of the confidence parameter β per intent θ via a maximum likelihood estimate:

$$\hat{\beta}_\theta = \arg\max_{\bar{\beta}} e^{-\bar{\beta}(C_\theta(\xi_{0\to t})+C_\theta(\xi^*_{t\to T})-C_\theta(\xi^*_{0\to T}))} \left(\frac{\bar{\beta}}{2\pi}\right)^{\frac{tk}{2}}, \quad (6)$$

where we drop the Hessians since they don't depend on β. Setting the derivative of the objective in Eq. (6) to zero and solving for β yields the following estimate:

$$\hat{\beta}^{MLE}_\theta = \frac{tk}{2(C_\theta(\xi_{0\to t}) + C_\theta(\xi^*_{t\to T}) - C_\theta(\xi^*_{0\to T}))} \quad . \quad (7)$$

Alternatively, we chose to add an exponential prior with parameter λ, $Exp(\lambda)$, on β to obtain a MAP estimate

$$\hat{\beta}^{MAP}_\theta = \frac{tk}{2(\lambda + C_\theta(\xi_{0\to t}) + C_\theta(\xi^*_{t\to T}) - C_\theta(\xi^*_{0\to T}))} \quad . \quad (8)$$

The denominators in equations 7 and 8 can be interpreted as the "suboptimality" of the observed partial trajectory $\xi_{0\to t}$ compared to the cost of the optimal trajectory for the particular θ, $C_\theta(\xi^*_{0\to T})$. Note that $\hat{\beta}_\theta$ is inversely proportional to the suboptimality divided by the number of time steps t that have passed. Intuitively, the user has more chances to be a suboptimal teleoperator as time goes on, so dividing for t corrects for the natural increase in suboptimality over time.

If this normalized suboptimality is low for an intent θ, then the person is close to a good trajectory for that intent and $\hat{\beta}_\theta$ will be high. Thus, a high $\hat{\beta}_\theta$ means that the person's input is well-explained by that intent. On the other hand, high suboptimality per time means the person is far from good trajectories, so θ's cost model C_θ does not explain the person's trajectory and $\hat{\beta}_\theta$ will be low.

D. Confidence-based Arbitration

Armed with a confidence estimate $\hat{\beta}_\theta$ for every $\theta \in \Theta$, the robot can predict the most likely one $\theta^* = \arg\max_{\theta\in\Theta} P(\theta \mid \xi_{0\to t}, \hat{\beta}_\theta)$ using Eq. (4). From here, one natural style of assistance is "policy blending" [2]. First the robot computes an optimal trajectory under the most likely intent, $\xi^* = \arg\min_\xi \sum_{x\in\xi} C^*_\theta(x)$, the first action of which is u^*. Then the robot combines u^* and $\mathcal{T}(a^t)$ using a blending parameter $\alpha \in [0, 1]$, resulting in the robot action $u^t = \alpha\mathcal{T}(a^t) + (1 - \alpha)u^*$. We also refer to α as the human's control authority.

Prior work proposes different ways to arbitrate between the robot and human actions by choosing α proportional to the robot's distance to the goal or to the probability of the most likely goal [2]. However, when using the probability $P(\theta^* \mid \xi)$, θ^* might look much better than the other intents, resulting in the robot wrongly assisting for θ^*. Distance-based arbitration ignores the full history of the user's input and can only accommodate simple intents.

Instead, we propose that the robot should use its confidence in the most likely intent, $\hat{\beta}_{\theta^*}$, estimated according to Sec. II-C, to control the strength of its arbitration:

$$u^t = \min(1, 1/\hat{\beta}_{\theta^*})\mathcal{T}(a^t) + (1 - \min(1, 1/\hat{\beta}_{\theta^*}))u^* \quad (9)$$

When $\hat{\beta}_{\theta^*}$ is high, i.e. the robot is confident that the predicted intent θ^* can explain the person's input, α is low, giving the robot more influence through its action u^*. When $\hat{\beta}_{\theta^*}$ is low, i.e. not even the most likely intent explains the person's input, α increases, giving the person's action a^t more authority.

E. Using Confidence for Lifelong Learning

Estimating the confidence $\hat{\beta}_\theta$ also lets the robot detect *misspecification* in Θ: if all estimated $\hat{\beta}_\theta$ for $\theta \in \Theta$ are below a threshold ϵ, the robot is missing the person's intent.

Once the robot has identified that its intent set is misspecified, it should ask the person to teach it. We represent the missing intent θ_ϕ as a neural network cost parameterized by ϕ and learn it via deep maximum entropy IRL [16]. The gradient of the IRL objective with respect to the cost parameters ϕ can be estimated by: $\nabla_\phi\mathcal{L} \approx \frac{1}{|\mathcal{D}^*|}\sum_{\tau\in\mathcal{D}^*}\nabla_\phi C_\phi(\tau) - \frac{1}{|\mathcal{D}^\phi|}\sum_{\tau\in\mathcal{D}^\phi}\nabla_\phi C_\phi(\tau)$. $\mathcal{D}^*$ are (noisy) demonstrations of the person executing the desired missing intent via direct teleoperation, and $\mathcal{D}^\phi$ are trajectories sampled from the C_ϕ induced near the optimal policy.

Once we have a new intent θ_ϕ, the robot updates its intent set $\Theta \leftarrow \Theta \cup \theta_\phi$. The next time the person needs assistance, the robot can perform confidence estimation, goal inference, and arbitration as before, using the new library of intents. While the complexity scales linearly with $|\Theta|$, planning can be parallelized across each intent.

Learned rewards fit naturally into our framework, allowing for a simple way to compare against the known intents. However, one could imagine adapting our method to the many other ways to learn an intent, from imitation learning [24], [25], to dynamic movement primitives [26]. For instance, if we parameterize intents via policies, we can derive a similar confidence metric based on probabilities of observed human actions under a stochastic policy, rather than costs.

III. Expert Case Study

In this section, we introduce three manipulation tasks and use expert data to analyze confidence estimation and assistance. We later put CASA's assistive capacity to test with non-experts in a user study in Sec. IV.

A. Experimental Setting

We conduct our experiments on the simulated 7-DoF JACO arm shown in Fig. 2. We use the pybullet interface [27] and teleoperate the robot via keypresses. We map 6 keys to bi-directional xyz movements of the robot's end-effector, and 2 keys for rotating it in both directions. We performed inference and confidence estimation twice per second.

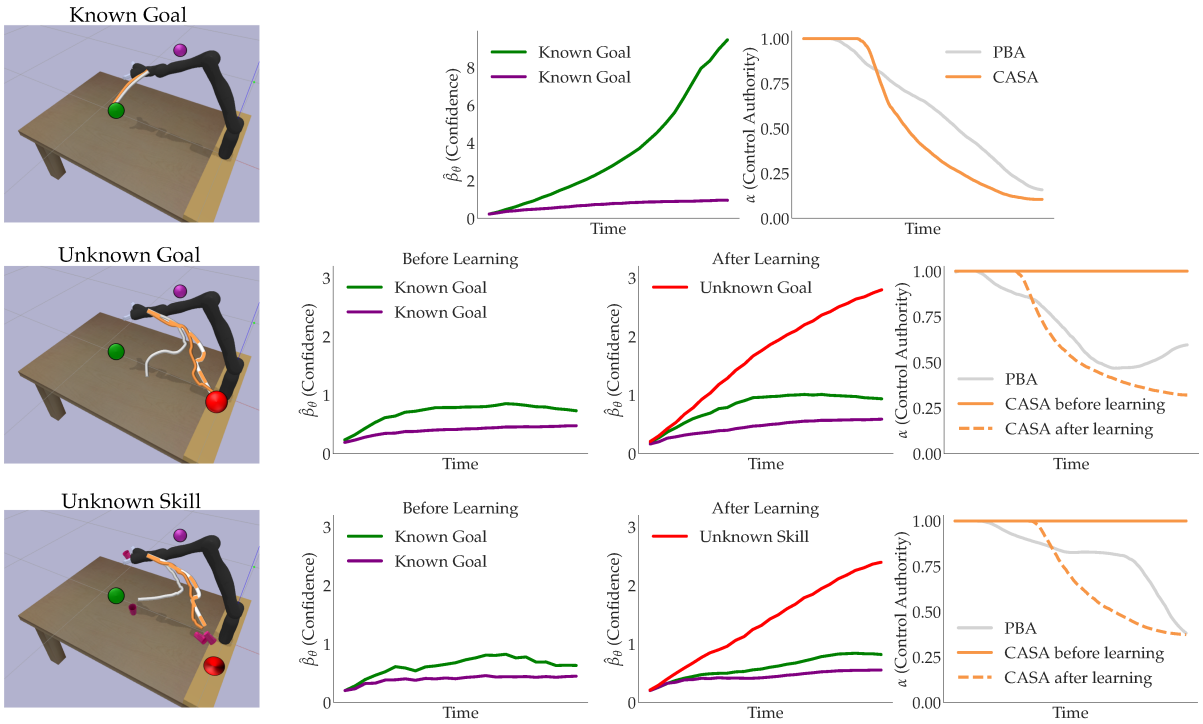

Fig. 2: Expert case study results. For each of three manipulation tasks, we compute confidence estimates before learning and, for the misspecified tasks (middle, bottom), we recompute the confidence estimates after learning. We also plot the strength of assistance before and after learning and compare to a policy blending baseline [2].

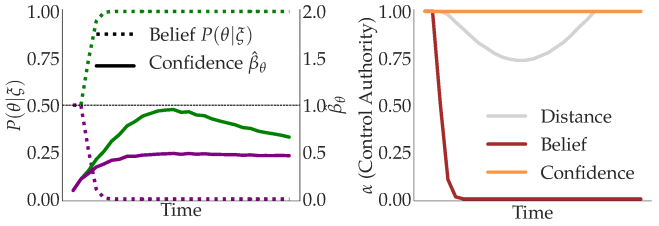

Fig. 3: Analysis of arbitration methods. After tracking an optimal trajectory for the Unknown Goal task, we show the robot's belief and confidence estimates for each known goal (left), as well as the α values under the distance, belief, and confidence-based arbitration schemes (right).

We test CASA on 3 different tasks. In the *Known Goal* task, we control for the well-specified setting: the robot must assist the user to move to the known green goal location in Fig. 2. In the other tasks, we test CASA's efficacy in the case of misspecification, where the user's desired intent is initially missing from the robot's known set Θ. In the second task, *Unknown Goal*, the person teleoperates the robot to the red goal which is unknown to the robot. Finally, in the third and most complicated task, *Unknown Skill*, the person tries to pour the cup contents at an unknown goal location.

For the Unknown Goal and Unknown Skill tasks, we first run CASA before being exposed to the new intent (CASA *before learning*). Detecting low confidence, the robot then asks for demonstrations and learns the missing intents via deep maximum entropy IRL as discussed in Sec. II-E. We then run teleoperation with CASA *after learning*, to assess the quality of robot assistance after learning the new intent.

B. Arbitration Method Comparison

We compare CASA to a policy blending assistance (PBA) baseline [2] that assumes $\beta = 1$ for all intents. PBA arbitrates with the distance d_{θ^*} to the predicted goal: $\alpha = \min(1, d_{\theta^*}/D)$, with D some threshold past which the robot does not assist. More sophisticated arbitration schemes use $P(\theta^* \mid \xi)$ or the full distribution $P(\theta \mid \xi)$, but they are much less robust to task misspecification. This is because when the user teleoperates for an unknown intent, $P(\xi \mid \theta)$ will be low for all known $\theta \in \Theta$; however, forming $P(\theta \mid \xi)$ requires normalizing over all known intents, after which $P(\theta^* \mid \xi)$ can still be high unless the user happened to operate in a way that appears equally unlikely under all known intents.

We analyzed this phenomenon by tracking a reference trajectory for the Unknown Goal task which moves optimally towards the unknown goal (see Fig. 2 for the task layout). We compared the performances of the distance and confidence arbitration methods, as well as a belief-based method which sets $\alpha = (P(\theta^* \mid \xi)|\Theta| - 1)/(|\Theta| - 1)$ (chosen so that $\alpha = 0$ when $P(\theta^* \mid \xi) = 1/|\Theta|$, $\alpha = 1$ when $P(\theta^* \mid \xi) = 1$). In Fig. 3, the confidence in each goal stays low enough that the robot would have left the user in full control; meanwhile, the relatively higher likelihood of one goal causes the belief $P(\theta^* \mid \xi)$ to quickly go to 1 and thus set the user's control authority to 0 under the belief-based arbitration scheme.

We examined one belief-based arbitration method here, but since $P(\theta^* \mid \xi)$ rapidly goes to 1, any other arbitration that is a function of the belief $P(\theta \mid \xi)$ would similarly try to assist for the wrong goal, motivating our choice of the simpler but more robust distance-based arbitration baseline.

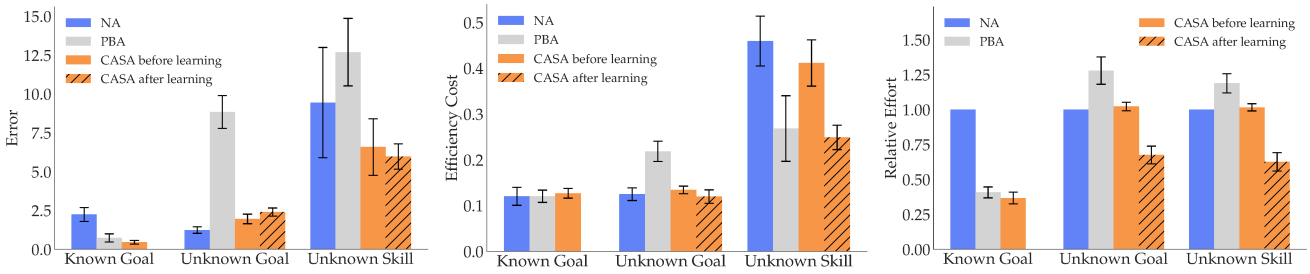

Fig. 4: Our user study objective metrics. For every task, we measured error with respect to an intended trajectory (left), smoothness of the executed trajectory (middle), and effort relative to direct teleoperation (right).

C. Well-specified Tasks

Fig. 2 (top) showcases the results of our experiment for the Known Goal task. Looking at the confidence plot, we see that $\hat{\beta}_\theta$ increases with time for the correct green goal, while it remains low for the alternate known purple goal. In the arbitration plot, as $\hat{\beta}_{\theta^*}$ increases, α gradually decreases, reflecting that the robot takes more control authority only as it becomes more confident that the person's intent is indeed θ^*. Similarly, since there is no misspecification, PBA arbitration steadily decreases the human's contribution to the final control. Both methods result in smooth trajectories which go to the correct goal location.

D. Misspecified Tasks

Our approach distinguishes itself in how it handles misspecified tasks. During the Unknown Goal task, in Fig. 2 (middle), CASA before learning estimates low $\hat{\beta}_\theta$ for both goals, since neither goal explains the person's motion moving towards the red goal. The estimated $\hat{\beta}_\theta$ is slightly higher for the green goal than for the purple one because it is closer to the user's input; however, neither are high enough to warrant an arbitration α below 1, and thus the robot receives no control. In Fig. 2 (bottom), we observe almost identical behavior before learning for the Unknown Skill task: the known intents do not match the user's behavior, and thus the user is given full control authority and completes the task.

This contrasts PBA, which, for both Unknown Goal and Unknown Skill, predicts the green goal as the intent. Since in both cases the user's desired trajectory passes near the green goal, PBA erroneously takes control and moves the user towards it, requiring the human to counteract the robot's controls to try to accomplish the task.

In the middle plots for each of the misspecified tasks, we observe for CASA after learning, the newly-learned intents receive confidence estimates which increase as the robot is able to observe the user, and thus CASA contributes more to the control as it becomes confident.

IV. User Study

We now present the results of our user study, testing how well our method can assist non-expert users.

A. Experimental Design

Due to the COVID-19 pandemic, we were unable to perform an in-person user study with a physical robot. Instead, as described in Sec. III, we replicated our lab set-up in a pybullet simulator [27] in which users can teleoperate a 7 DoF JACO robotic arm using keyboard inputs (Fig. 2).

We split the study into four phases: (1) familiarization, (2) no misspecification, (3) misspecification before learning, and (4) misspecification after learning. First, we introduce the user to the simulation interface by asking them to perform a familiarization task. In the next phase, we tested the Known Goal task. In the third phase, we tested the two misspecified tasks, Unknown Goal and Unknown Skill, then asked participants to provide 5 demonstrations for each intent. Finally, in the fourth phase, we retested the misspecified tasks using cost functions learned from the demonstrations.

Independent Variables: For each experiment, we manipulate the *assistance method* with three levels: no assistance (NA), policy blending assistance (PBA) [2], and Confidence-Aware Shared Autonomy (CASA). For Unknown Goal and Unknown Skill, we compared our method before and after learning new intents against the NA and PBA baselines.

Dependent Measures: Before each task, we displayed an exemplary reference trajectory to help participants understand their objective. As such, for our objective metrics, we measured *Error* as the sum of squared differences between the intended and executed trajectories, *Efficiency Cost* as the sum of squared velocities across the executed trajectory, and *Effort* as the number of keys pressed. To assess the users' interaction experience, we administered a subjective 7-point Likert scale survey, asking the participants three questions: (1) if they felt the robot understood how they wanted the task done, (2) if the robot made the interaction more effortless, and (3) if the assistance provided was useful.

Participants: We used a within-subjects design and counterbalanced the order of the assistance methods. We recruited 11 users (10 male, aged 20-30) from the campus community, most of whom had technical background.

Hypotheses:

H1: If there is no misspecification, assisting with CASA is not inferior to assisting with PBA, and is superior to NA.

H2: If there is misspecification, assisting with CASA before learning is more accurate, efficient, and effortless than with PBA and not inferior to NA.

H3: If there is misspecification, assisting with CASA after learning is more accurate, efficient, and effortless than NA.

H4: If there is misspecification, participants will believe the robot understood what they want, feel less interaction effort,

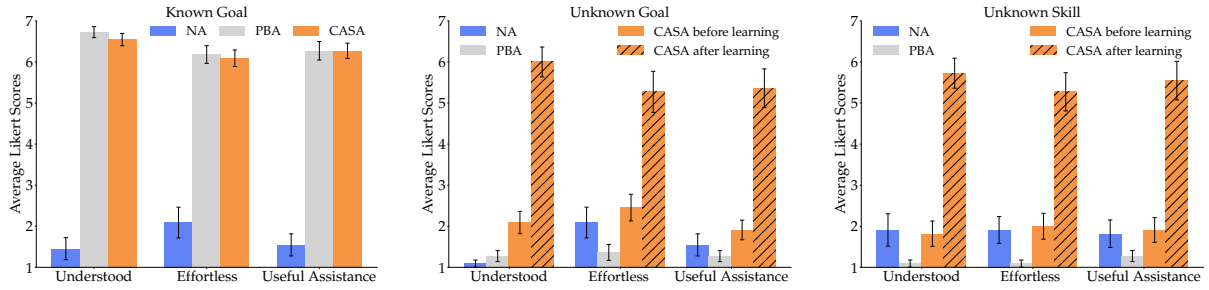

Fig. 5: Subjective user study results. When there is no misspecification (left), our method is not inferior to PBA, whereas when there is misspecification (center, right), the participants prefer our method after learning a new intent.

and find the assistance more useful with CASA after learning than with any other baseline.

B. Analysis

Objective. Fig. 4 summarizes our main findings. For Known Goal, which is well-specified, CASA does no worse than PBA and better that NA in terms of relative effort and error. We confirmed this by running an ANOVA, finding a significant main effect for the method ($F(2,30) = 104.93, p < .0001$ for effort; $F(2,30) = 8.93, p = .0009$ for error). In post-hoc testing, a Tukey HSD test revealed that CASA is significantly better than NA ($p < .0001$ for effort, $p = .0013$ for error). We also performed a non-inferiority test [28], and obtained that CASA is non-inferior to PBA within a margin of 0.065 for effort, 0.025 for efficiency, and 0.26 for error. These findings are in line with H1 and were expected, since the robot should have no problem handling known intents.

For the two misspecified tasks, we first ran an ANOVA with the method (CASA before learning, NA, and PBA) as a factor, and the task as a covariate, and found a significant main effect ($F(2,62) = 11.8255, p < .0001$ for effort; $F(2,62) = 6.119, p = .0038$ for error). A Tukey HSD revealed that CASA is significantly better than PBA ($p = .0005$ for effort, $p = .005$ for error). We also ran a non-inferiority test, and obtained that CASA is non-inferior to NA within a margin of 0.035 for effort, 0.02 for efficiency, and 1.4 for error for Unknown Goal, and 0.03 for effort, 0.09 for efficiency, and 4.5 for error for Unknown Skill. For both unknown tasks, CASA before learning is essentially indistinguishable from NA since a low $\hat{\beta}_{\theta^*}$ would make the robot rely on direct teleoperation. Both the figure and our statistical tests confirm H2, which speaks for the consequences of confidently assisting for the wrong intent.

For efficiency cost, we did not find an effect, possibly because Fig. 4 shows that PBA is more efficient for the Unknown Skill task than other methods. Anecdotally, PBA forced users to an incorrect goal thus preventing them from pouring, which explains the lower efficiency cost. By having a high arbitration for the wrong intent, PBA can cause a smooth trajectory, since it lowers the control authority of the possibly-noisy human inputs. However, this trajectory does not accomplish the task. When running an ANOVA for each of the tasks separately, we found a significant main effect for the method for Unknown Goal ($F(2,30) = 9.66, p = .0006$), and a post-hoc Tukey HSD revealed CASA is significantly better than PBA ($p = .0032$), further confirming H2.

Lastly, we looked at the performance with CASA after learning the new intents. For Unknown Goal, a simple task, the figure shows that CASA after learning doesn't improve efficiency and error, but it does reduce relative effort when compared to NA. For Unknown Skill, a more complex task, CASA after learning outperforms NA. This is confirmed by an ANOVA with the method (NA, CASA after learning) as the factor, where we found a significant main effect ($F(1,41) = 53.60, p < .0001$ for effort; $F(1,641) = 8.6184, p = .0054$ for efficiency cost), supporting H3.

Subjective. We show the average Likert survey scores for each task in Fig. 5. In line with H1, for the Known Goal task, users thought the robot under both PBA and CASA had a good understanding of how they wanted the task to be done, made the interaction more effortless, and provided useful assistance. The results are in stark contrast to NA, which scores low on all those metrics. For Unknown Goal and Unknown Skill, all methods fare poorly on all questions except for CASA after learning, supporting H4.

V. Conclusion

In this paper, we formalized a confidence-aware shared autonomy process where the robot can adjust its assistance based on how confident it is in its prediction of the human intent. We introduced an approximate solution for estimating this confidence, and demonstrated its effectiveness in adjusting arbitration when the robot's intent set is misspecified and enabling continual learning of new intents.

While our confidence estimates tolerated some degree of suboptimal user control, an extremely noisy operator attempting a known intent might instead appear to be performing a novel intent. Moreover, due to COVID, we ran our experiments in a simulator, which does not replicate the difficulty inherent in teleoperating a real manipulator via a joystick interface. Despite these limitations, we are encouraged to see robots have a more principled and robust way to arbitrate shared autonomy, as well as decide when they need to learn more to be better teammates. We look forward to applications of our confidence-based ideas beyond manipulation robots, to semi-autonomous vehicles, quadcopter control, or any other shared autonomy scenarios.

REFERENCES

[1] P. Aigner and B. McCarragher, "Human integration into robot control utilising potential fields," in *Proceedings of International Conference on Robotics and Automation*, vol. 1. IEEE, 1997, pp. 291–296.

[2] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013. [Online]. Available: https://doi.org/10.1177/0278364913490324

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

[4] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A Review of Intent Detection, Arbitration, and Communication Aspects of Shared Control for Physical Human–Robot Interaction," *Applied Mechanics Reviews*, vol. 70, no. 1, 02 2018, 010804. [Online]. Available: https://doi.org/10.1115/1.4039145

[5] R. C. Goertz, "Manipulators used for handling radioactive materials," *Human factors in technology*, pp. 425–443, 1963.

[6] F. Abi-Farraj, C. Pacchierotti, and P. R. Giordano, "User evaluation of a haptic-enabled shared-control approach for robotic telemanipulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–9.

[7] D. P. Losey, K. Srinivasan, A. Mandlekar, A. Garg, and D. Sadigh, "Controlling assistive robots with learned latent actions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 378–384.

[8] Q. Li, W. Chen, and J. Wang, "Dynamic shared control for human-wheelchair cooperation," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4278–4283.

[9] A. Erdogan and B. D. Argall, "The effect of robotic wheelchair control paradigm and interface on user performance, effort and preference: an experimental assessment," *Robotics and Autonomous Systems*, vol. 94, pp. 282–297, 2017.

[10] D. S. Brown, S.-Y. Jung, and M. A. Goodrich, "Balancing human and inter-agent influences for shared control of bio-inspired collectives," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 4123–4128.

[11] J. W. Crandall, N. Anderson, C. Ashcraft, J. Grosh, J. Henderson, J. McClellan, A. Neupane, and M. A. Goodrich, "Human-swarm interaction as shared control: Achieving flexible fault-tolerant systems," in *International Conference on Engineering Psychology and Cognitive Ergonomics*. Springer, 2017, pp. 266–284.

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

[13] K. Muelling, A. Venkatraman, J.-S. Valois, J. E. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell, "Autonomy infused teleoperation with application to brain computer interface controlled manipulation," *Autonomous Robots*, vol. 41, no. 6, pp. 1401–1422, 2017.

[14] C. Pérez-D'Arpino and J. A. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6175–6182.

[15] A. Bobu, A. Bajcsy, J. F. Fisac, S. Deglurkar, and A. D. Dragan, "Quantifying hypothesis space misspecification in learning from human–robot demonstrations and physical corrections," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 835–854, 2020.

[16] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International conference on machine learning*, 2016, pp. 49–58.

[17] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, ser. AAAI'08. AAAI Press, 2008, pp. 1433–1438. [Online]. Available: http://dl.acm.org/citation.cfm?id=1620270.1620297

[18] A. Ng and S. Russell, "Algorithms for inverse reinforcement learning," *International Conference on Machine Learning (ICML)*, vol. 0, pp. 663–670, 2000. [Online]. Available: http://www-cs.stanford.edu/people/ang/papers/icml00-irl.pdf

[19] C. Baker, J. B Tenenbaum, and R. R Saxe, "Goal inference as inverse planning," 01 2007.

[20] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton University Press Princeton, NJ, 1945.

[21] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin, "Confidence-aware motion prediction for real-time collision avoidance," *International Journal of Robotics Research*, 2019.

[22] J. F. Fisac, A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, S. Wang, C. J. Tomlin, and A. D. Dragan, "Probabilistically safe robot planning with confidence-based human predictions," *Robotics: Science and Systems (RSS)*, 2018.

[23] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization."

[24] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4565–4573. [Online]. Available: http://papers.nips.cc/paper/6391-generative-adversarial-imitation-learning.pdf

[25] S. Reddy, A. D. Dragan, and S. Levine, "SQIL: imitation learning via reinforcement learning with sparse rewards," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=S1xKd24twB

[26] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2616–2624. [Online]. Available: http://papers.nips.cc/paper/5177-probabilistic-movement-primitives.pdf

[27] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.

[28] E. Lesaffre, "Superiority, equivalence, and non-inferiority trials," *Bulletin of the NYU hospital for joint diseases*, vol. 66, no. 2, p. 150—154, 2008. [Online]. Available: http://europepmc.org/abstract/MED/18537788