Quantifying Hypothesis Space Misspecification in Learning from Human-Robot Demonstrations and Physical Corrections

Andreea Bobu, *Student Member, IEEE,* Andrea Bajcsy, *Student Member, IEEE,* Jaime F. Fisac, *Student Member, IEEE,* Sampada Deglurkar, and Anca D. Dragan *Member, IEEE*

(Invited Paper)

***Abstract*—Human input has enabled autonomous systems to improve their capabilities and achieve complex behaviors that are otherwise challenging to generate automatically. Recent work focuses on how robots can use such input — like demonstrations or corrections — to learn intended objectives. These techniques assume that the human's desired objective already exists within the robot's hypothesis space. In reality, this assumption is often inaccurate: there will always be situations where the person might care about aspects of the task that the robot does not know about. Without this knowledge, the robot cannot infer the correct objective. Hence, when the robot's hypothesis space is *misspecified*, even methods that keep track of uncertainty over the objective fail because they reason about which hypothesis might be correct, and not whether *any* of the hypotheses are correct. In this paper, we posit that the robot should reason explicitly about how well it can explain human inputs given its hypothesis space and use that *situational confidence* to inform how it should incorporate human input. We demonstrate our method on a 7 degree-of-freedom robot manipulator in learning from two important types of human input: demonstrations of motion planning tasks, and physical corrections during the robot's task execution.**

***Index Terms*—Bayesian inference, physical human-robot interaction, learning from demonstration, inverse reinforcement learning.**

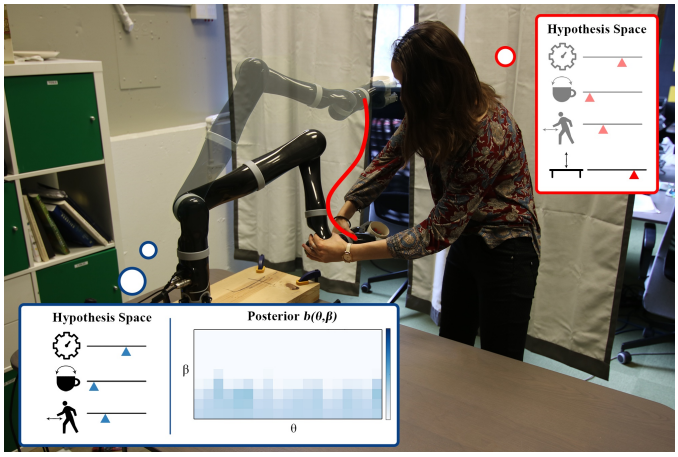


Fig. 1: A household robotics scenario where the person physically interacts with the robot. The person prefers the robot to keep cups closer to the table, but accounting for the table (outside of collisions) is not in the robot's hypothesis space for what the person might care about. Thus, the robot's internal situational confidence, β, about what the human input means is low for all hypotheses θ.

I. Introduction

AUTONOMOUS systems are increasingly interfacing and collaborating with humans in a variety of contexts, such as semi-autonomous driving, automated control schemes on airplanes, or household robots working in close proximity with people. While the improving capabilities of robotic systems are opening the door to new application domains, the substantially greater complexity and interactivity of these settings makes it challenging for system designers to account for all relevant operating conditions and requirements ahead of time. For example, a household robot designer may not know how an end-user would like the robot to interact with the personal possessions in the user's home.

In situations like these, it can be beneficial for the robot to utilize human input as guidance on the desired behavior. In fact, human input has enabled researchers and engineers to program advanced behaviors that would have otherwise been extremely challenging to specify. Helicopter acrobatics [1], aggressive automated car maneuvers [2], and indoor navigation [3] are three cases that exemplify the benefit of using human input for guiding robot behavior.

In order to utilize human input, system designers typically equip robots with a representation of possible objectives that the human *could* care about. These representations can range from quadratic cost models [4] to complex temporal logic specifications [5] to neural networks [6]. However, anticipating all motivations for human input and specifying a complete model is challenging. Consider Fig. 1 where a human is attempting to change the robot's behavior in order to make it consistently stay close to the table, but the robot's model of what the human might care about does not include distances to the table. By choosing a class of functions, the system designer implicitly assumes that what the human wants (and is giving input about) can be represented via a member of that class. Unfortunately, when this assumption breaks, the

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Andreea Bobu, UC Berkeley, Berkeley, CA, 94709 USA
e-mail: abobu@eecs.berkeley.edu
A. Bajcsy, J. F. Fisac, S. Deglurkar, and A. D. Dragan are with the EECS Department at University of California, Berkeley.
Manuscript received April 20, 2019; revised October 13, 2019.

system can misinterpret human guidance, perform unexpected or undesired behavior, and degrade in overall performance.

Two approaches to mitigating this problem could be to either start with a more complex objective space or to continuously increase its complexity given more data. Unfortunately, even complex models are not guaranteed to encompass all possibilities and re-computing the best objective space based on human data faces the threat of overfitting to the most recent observations. In contrast, we argue the robot should be able to *understand when it cannot understand the input*. For example, if the end-user in the home is trying to guide the robot to handle fragile objects with care but the system does not posses a model of fragility, the robot should deduce that this input cannot be well explained by any of its given hypotheses.

In this work, we formalize how autonomous systems can explicitly reason about how well they can explain given human inputs. To do this, we observe that if a human input appears unlikely with respect to all possible hypotheses, then the robot's model is misspecified. We build on previous work centered around this observation to propose a Bayesian inference framework focused on inferring both model parameters, and their corresponding *situational confidence*. If the robot is in situations like Fig. 1 where none of the hypotheses explain the human's input well, then the situational confidence will be low for all hypotheses, indicating that the robot's model is not sufficiently rich to understand the human's input. However, when the robot's model is well specified, our framework does not impede the robot from inferring the correct task objectives — in fact, the situational confidence will be high, providing an indicator of how well the system can understand the objective.

We illustrate the utility of situational confidence estimation in quantifying objective space misspecification for two types of human input: demonstrations and corrections. Our contributions in this work are:

1) we introduce a general framework for quantifying objective space misspecification when the human and the robot are acting on the same dynamical system;
2) we showcase the framework for learning from demonstrations using user demonstration data for an arm motion planning task;
3) we showcase the framework for learning from physical corrections by deriving an algorithm for online (close to real-time) inference and testing it in a user study.

We note that this work is an extension of [7], which was originally presented at the Conference on Robot Learning, 2018. We build on this work by introducing a general framework for quantifying objective space misspecification, and instantiating it in a new type of human input: learning from demonstrations. Not only are demonstrations the most widely used type of input for learning objective functions, but the applicability across two input types suggests that the approach could be adapted more broadly to more types of human feedback.

The remainder of this paper is organized as follows: Section II places this work in the context of existing literature on robots learning from humans and model confidence estimation. Section III frames the confidence estimation problem more formally for scenarios where the human and robot operate on the same dynamical system. Section IV directly instantiates the framework in Section III for the case of learning from demonstrations. Section V presents a derivation of approximations of the general formalism for tractable online inference from human corrections. Section VI showcases our proposed approach in several case studies where the robot's hypothesis space cannot or only partially explain the human's input. Section VII presents the results of a user study of our approach as applied to a 7-DoF robotic manipulator learning from human participants. Section VIII concludes with a discussion of some of the limitations of our work, as well as suggestions for future research directions.

Overall, we think that the ability to detect misspecification when learning objectives from human input will become increasingly important as robotics capability advances and we will want end-users to customize how the robot behaves. Our work takes a step in this direction by enabling robots to detect when none of the hypotheses they have explain the user input, and our experiments show promising results. Of course, there are still limitations to this. One limitation is in the experiments themselves, which are only for motion planning tasks with low-dimensional hypothesis spaces. A more fundamental limitation is that there will still be cases when the person wants something outside the robot's hypothesis space, but the robot can nonetheless explain their current input relatively well with what it has access to, thus confusing misspecification for slight noise in the human input. This will especially be the case as the hypothesis space is more expressive, and can only be solved by the robot receiving a lot more human input: each might be explainable by some hypothesis, but eventually no hypothesis can explain all input. More work is needed in studying how to query for diverse human input, as well as how to convey what the robot has learned back to the person, and in general how to have a true collaborative interaction to detect and resolve misspecification in the objective space.

II. Related Work

We group prior work into three main categories: enabling robots to learn from human input, doing so while leveraging uncertainty, and estimating model confidence.

A. Robots learning from humans

The programming of robots through direct human interaction is a well-established paradigm. Human input can be given to the robot in a variety of forms, from teleoperation of the robot by a user to kinesthetic teaching [8].

In such interaction paradigms, the robot aims to infer a *cost function* or *policy* that best describes the examples that it has received. New avenues of research focus on learning such robot objectives from human input through demonstrations [9], [10], teleoperation data [11], corrections [12], [13], comparisons [14], examples of what constitutes a goal [15], or even specified proxy objectives [16]. In this paper, we focus on learning from two of such types of human input – demonstrations and physical corrections – although we stress that the principles outlined in our formalism are more general and could be applied to the other interaction modes mentioned.

One approach to learning behaviors from human inputs is inverse reinforcement learning (IRL). In classical IRL, the robot receives complete optimal *demonstrations* of how to perform a task, and the robot learns the human's cost function from these observations [10], [17], [18]. In this paradigm, it is typically assumed that the expert is trying to optimize an unknown cost function. The robot uses the observations of the human's behavior to recover the underlying objective.

Another useful form of human input are *corrections*: here, the robot performs the task according to how it was programmed and the user corrects aspects of the task to better match their preferences. From these sparse interactions, the robot also performs cost function inference to improve performance during the next task iteration [19]–[21]. Examples of learning from corrections have been explored in offline [12], [22] and online settings [13], [23], [24], [25].

Although powerful, the aforementioned IRL works assume that the human expert provides optimal demonstrations, which is often an unrealistic assumption. Real human input, especially during interaction with high degree-of-freedom systems like robotic manipulators, is noisy and sub-optimal. Second, much of the corrections literature has focused on estimates of the human's objectives. However, in practice, even the most likely estimate might not be a very likely one. Thus, in both domains, we stress that it is important to maintain the uncertainty over the estimated objectives.

B. Uncertainty in robot learning

Rather than estimating a single objective, some learning methods maintain an entire probability distribution over what the objective might be [16], [26]–[28]. This not only enables the robot to leverage a prior, but also to then generate its behavior in a way that is mindful of the entire distribution, rather than just using the the maximum likelihood estimator.

Bayesian IRL [28] treats demonstrations as evidence about the objective, and does a Bayesian belief update on a prior distribution. Inverse Reward Desing [16] treats the objective a designer specified for a particular set of environments (a "proxy" objective) as evidence about the true desired objective, again obtaining a full distribution over what the designer might actually want. The intuition is that this observed proxy objective (that may be misspecified) incentivizes behavior that is approximately optimal with respect to the true objective.

Lastly, specifically for input as physical corrections, [27] reasons over the uncertainty of the estimated human preferences through the means of a Kalman filter. The method maintains a mean estimate and a covariance of this estimate as a measure of confidence. These are used in planning the robot's trajectory such that it optimizes for features it is confident about, while avoiding features it is uncertain about.

Although they maintain a full distribution, these works still assume that what the human wants is in the robot's objective space. We argue that this is not necessarily a realistic assumption, and later showcase some consequences that arise when it is not true. When the robot's hypothesis space is misspecified, even when maintaining uncertainty over the objective, state-of-the-art methods interpret human input as evidence about which hypothesis is correct, rather than considering whether any hypothesis is correct. In this work, we focus on the latter.

C. Situational confidence estimation

Some recent works are studying how to enable robots to understand that their models cannot explain human input well [29]–[31]. The authors in [30], [31] employ a noisily-optimal model of human pedestrian motion when the human and the robot operate on separate dynamical systems (and have separate objective functions). The paper introduces the notion of model confidence estimation and uses the apparent likelihood of the human's choice of actions to adjust the confidence in predictions about their behavior.

This work draws inspiration from the notion of model confidence estimation, generalizing it to the setting of inferring what the robot's objective ought to be. Instead of focusing on misspecification of a discrete set of physical goal locations for pedestrian navigation, here we study misspecification of a relatively complex set of possible robot objectives in motion planning tasks. As a result of focusing on robot objectives, we also study a different form of human input – that is, input in the context of operating on the same dynamical system, such as full task demonstrations and physical corrections.

III. Problem Formulation and Approach

We consider a robot R operating in the presence of a human H whom it seeks to assist in the execution of some task. In the most general setting, the robot and the human are both able to affect the evolution of the state $x \in \mathbb{R}^n$ over time through their respective control inputs:

$$x^{t+1} = f\left(x^t, u_R^t, u_H^t\right) \ , \tag{1}$$

with $u_R \in \mathcal{U}_R$ and $u_H \in \mathcal{U}_H$, where $\mathcal{U}_i$ ($i \in \{H, R\}$) are compact sets. We assume that the human has some consistent preference ordering between different state trajectories and input signals, which could in principle be expressed through a cost function of the form

$$C^*(\mathbf{x}, \mathbf{u}_R, \mathbf{u}_H) \tag{2}$$

where the state trajectory is $\mathbf{x} = [x^0, x^1, \dots, x^T] \in \mathbb{R}^{n(T+1)}$, the robot's control input is $\mathbf{u}_R = [u_R^0, u_R^1, \dots, u_R^T] \in \mathbb{R}^{n(T+1)}$, and the human's is $\mathbf{u}_H = [u_H^0, u_H^1, \dots, u_H^T] \in \mathbb{R}^{n(T+1)}$.[1] Note that this hypothesized cost function C^* can be quite general, encoding an arbitrary preference ordering. However, the robot does not in general have access to the human's preferences C^*, and must instead attempt to infer and represent them tractably.

In order to do this, the robot can typically reason over a parametrized approximation of the cost function, which introduces an inductive bias, making inference tractable at the cost of limiting expressiveness: in some cases, the chosen set of parametric functions may fail to encode preferences that would explain the human's behavior with sufficient accuracy. In this work, we will denote by C_θ the cost function induced

[1] For deterministic dynamics (1), having $x^0, \mathbf{u}_R$ and $\mathbf{u}_H$ is enough to fully specify the entire state trajectory $\mathbf{x}$. In this case, the cost function could be rewritten as $C^*(x^0, \mathbf{u}_R, \mathbf{u}_H)$ by implicitly encoding (1). For clarity, we use the more general form in (2) and make the dependence explicit where needed.

by parameters $\theta \in \Theta$, and the robot seeks to estimate the human's preferred θ from her control inputs $\mathbf{u}_H$.

In a general setting, since the state trajectory $\mathbf{x}$ is determined not only by the human's actions $\mathbf{u}_H$ but also the robot's $\mathbf{u}_R$, the human would need to reason about how the robot will respond to her decisions. This requires analyzing the interaction in a game-theoretic framework [32], [33], which will not be the object of this work. Instead, we focus on common interaction scenarios in which the robot can approximately assume that the human does not explicitly account for the coupled mutual influence between both agents' decisions. This happens frequently if the human is either providing a demonstration for the robot or intervening to correct the robot's default behavior. In these settings, the typical assumption is that the human has all necessary information about the robot's control input $\mathbf{u}_R$ before deciding on her own $\mathbf{u}_H$.

Thus, given observations of the human input $\mathbf{u}_H$ from an initial state x^0, the robot needs to draw inferences on the cost parameter θ:

$$P(\theta \mid x^0, \mathbf{u}_R, \mathbf{u}_H) = \frac{P(\mathbf{u}_H \mid x^0, \mathbf{u}_R; \theta) P(\theta)}{\int_{\bar{\theta}} P(\mathbf{u}_H \mid x^0, \mathbf{u}_R; \bar{\theta}) P(\bar{\theta}) d\bar{\theta}} \ , \quad (3)$$

where $P(\mathbf{u}_H \mid x^0, \mathbf{u}_R; \theta)$ characterizes how the robot expects the human's input to be informed by her preferences, conditioned on the initial state and the robot's expected controls.

For example, if the human were assumed to act optimally, this model would place all probability on the set of optimal states and actions with respect to the cost C_θ. Of course, this would be an unreasonably strong assumption given that the robot's parametrized cost constitutes a best effort to approximate the human's preferences. Instead, a useful modeling choice can be to characterize the human as being more *likely* to take actions that are well-aligned with her preferences.

One such model is inspired by the Boltzmann energy-based model satisfying the maximum entropy principle [34]. Following its adaptations as a model of human decision-making in [13], [35], [36], we model the human as a noisily-optimal agent that tends to choose control inputs that approximately minimize the modeled cost:

$$P(\mathbf{u}_H \mid x^0, \mathbf{u}_R; \theta, \beta) = \frac{e^{-\beta C_\theta \left(\mathbf{x}(\cdot; x^0, \mathbf{u}_R, \mathbf{u}_H), \mathbf{u}_R, \mathbf{u}_H\right)}}{\int_{\bar{\mathbf{u}}_H} e^{-\beta C_\theta \left(\mathbf{x}(\cdot; x^0, \mathbf{u}_R, \bar{\mathbf{u}}_H), \mathbf{u}_R, \bar{\mathbf{u}}_H\right)} d\bar{\mathbf{u}}_H}. \quad (4)$$

In this model, the inverse temperature coefficient $\beta \in [0, \infty)$ determines the degree to which the robot expects to observe human actions that are consistent with the cost model.

The goal is to detect when the robot does not have a rich enough hypothesis space, i.e. when C^* lies far outside of any C_θ. We call this problem *objective space misspecification*. Rather than only interpreting human input as evidence about *which* hypothesis is correct, we additionally focus on considering whether *any* hypothesis is correct. It is thus crucial that the robot can quantify the extent to which any parameter value $\theta \in \Theta$ can correctly explain the observed human input.

A. Situational confidence estimation

The key to our approach goes back to the inverse temperature parameter β in (4). Typically, β is a fixed term, encoding the degree to which the robot expects to observe human actions that are optimal. Setting it to 0 models a randomly-acting human, while setting it to ∞ models a perfectly optimal human. However, the possibility of objective space misspecification brings fixing β into question: when the space is correctly specified, we would expect the human actions to indeed be somewhat close to optimal; but when the space is misspecified, *we should expect the actions to be far from optimal for any* θ. Thus, rather than treating β as a fixed term, we build on the work in [30], [31] and explicitly reason over β as an additional inference parameter along with θ. Since β directly impacts the entropy of the human's decision model, it can be used as an effective and computationally efficient measure of the robot's confidence in its parametric interpretation of the human's preference: we say that the robot is assessing its *situational confidence* for the inference task at hand.

Thus, the robot maintains a joint Bayesian belief $b(\theta, \beta)$. For each new measurement of $\mathbf{u}_H$ given $x^0, \mathbf{u}_R$, this belief is updated as:

$$b'(\theta, \beta) = \frac{P(\mathbf{u}_H \mid x^0, \mathbf{u}_R; \theta, \beta) b(\theta, \beta)}{\int_{\bar{\theta}, \bar{\beta}} P(\mathbf{u}_H \mid x^0, \mathbf{u}_R; \bar{\theta}, \bar{\beta}) b(\bar{\theta}, \bar{\beta}) d\bar{\theta} d\bar{\beta}} \ , \quad (5)$$

where $b'(\theta, \beta) = P(\theta, \beta \mid x^0, \mathbf{u}_R, \mathbf{u}_H)$.

This inference can be seen as analogous to performing Bayesian Inverse Reinforcement Learning [28] with the Maximum Entropy Inverse Optimal Control [37] observation model, where we maintain the full belief instead of just the maximum likelihood estimate, and we explicitly reason over the additional scaling parameter β. By actively performing inference over β, the robot can gain insight into the reliability of its human model in light of new evidence.

1) Context-dependent usage of situational confidence: How this insight should be used is dependent on the context of the robot's operation. Here, we provide some examples of how situational confidence can be integrated into various human-robot interaction scenarios and robot motion planners.

In collaborative settings where the human and robot are accomplishing a task together (e.g. manipulating an object together), it may be desirable for the robot to stop and ask for clarification from the human whenever sufficient probability mass indicates low confidence:

$$\forall \theta \in \Theta, \arg\max_{\beta} b'(\beta \mid \theta) < \epsilon \ . \quad (6)$$

That is, for a predefined threshold ϵ, if all hypotheses have the most mass on βs lower than ϵ, the robot can raise a flag.

In assistive applications, where the robot is carrying out a task in close physical proximity to the human, the robot may receive intermittent human input to correct it's task performance. In such scenarios, it may be appropriate for the robot to simply dismiss human corrections that it cannot explain in terms of modeled preference parameters and carry on with its pre-defined task. That is, when a human input results in a $b'(\theta, \beta)$ that satisfies (6), the input gets discarded.

Situational confidence could also be leveraged by robot motion planners that excel at decision making under uncertainty. Here, the robot may use its joint posterior belief $b'(\theta, \beta)$ to make goal-driven decisions in the presence of the human. To

this end, the coupling between the inference problem and the robot's planning problem can be viewed as a partially observable Markov decision process (POMDP), where the hidden parts of the state are the cost parameter θ and the situational confidence β, the robot receives observations about them via human actions $\mathbf{u}_H$, it takes actions $\mathbf{u}_R$, and it optimizes an unknown parametrized cost C_θ. Our problem is, thus, akin to identifying misspecification in the state space of the POMDP. However, inference and planning in such spaces requires solving the full POMDP, which is computationally intractable for large, real-world problems [38].

Alternative, less computationally demanding motion planning approaches are also amenable to our framework, where the robot plans to minimize the expected cost for the human given its current belief, by marginalizing over β:

$$\min_{\mathbf{u}_R} \mathop{\mathbb{E}}_{\theta\sim b} \left[C_\theta(\mathbf{x}, \mathbf{u}_R, \mathbf{u}_H)\right] \ , \tag{7}$$

for an expected human input $\mathbf{u}_H$ that will typically be $\mathbf{0}$ if the robot is attempting to successfully perform the task without the need for active human intervention. To understand the implication (7) has as a function of the inference over β, we need to understand the posterior belief marginalized over β that we are taking the expectation over. At one extreme, if for all θs the conditional distribution $b'(\beta \mid \theta)$ puts all probability mass on $\beta = 0$ (i.e. input poorly explained), since $P(\mathbf{u}_H \mid x^0, \mathbf{u}_R; \theta, \beta = 0)$ is the same for all θs, the robot will obtain a posterior for θ that is equal to the prior. The optimization above becomes the same as optimizing using the robot's prior, i.e. the robot ignores the human input. At the other extreme, if there is one θ that perfectly explains the input and all others do not, the posterior will put all probability mass on that θ, and the robot will switch to optimizing it.

The objective expectation may also be appropriately weighted by the robot's situational confidence for each θ:

$$\min_{\mathbf{u}_R} \mathop{\mathbb{E}}_{\theta,\beta\sim b} \left[\beta C_\theta(\mathbf{x}, \mathbf{u}_R, \mathbf{u}_H)\right] \ , \tag{8}$$

which leads to the robot prioritizing those components of the task about which it is most certain.

In Sections IV and V we discuss some of these possibilities in the context of learning from demonstrations and corrections.

B. Cost representation through basis functions

One way to approximate the infinite-dimensional space of possible cost functions using a finite number of parameters is the use of a finite family of basis functions Φ_i [18]. This family can be seen as a truncation of an infinite collection of basis functions spanning the full function space. Parametric approximations C_θ of the cost function C^* then have the form

$$C_\theta(\mathbf{x}, \mathbf{u}_R, \mathbf{u}_H) = \sum_{i=1}^{d} \theta^i \Phi_i(\mathbf{x}, \mathbf{u}_R, \mathbf{u}_H) = \theta^T \Phi(\mathbf{x}, \mathbf{u}_R, \mathbf{u}_H) \ . \tag{9}$$

Consistent with classical utility theories [35], we further assume that the human's preferences can be approximated through a cumulative return over time, rewriting (9) as

$$C_\theta(\mathbf{x}, \mathbf{u}_R, \mathbf{u}_H) = \sum_{i=1}^{d} \theta^i \sum_{t=0}^{T} \phi_i(x^t, u_R^t, u_H^t) \ , \tag{10}$$

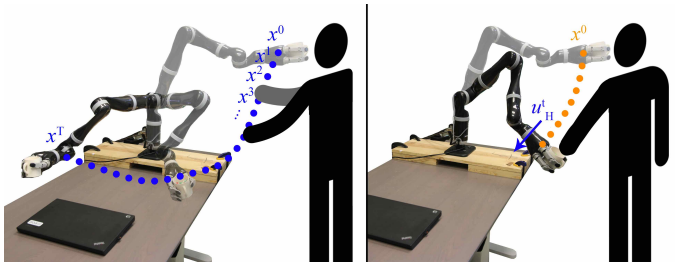


Fig. 2: (Left) Visual example of a full human-provided demonstration $\mathbf{x}$. (Right) Visual example of a human physical correction u_H^t onto the robot's current trajectory $\mathbf{x}$.

where $\phi_i : \mathbb{R}^n \times \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ are fixed, pre-specified, bounded real-valued basis functions, θ is the unknown parameter that the robot is trying to fit according to the human's preferences, and d is the dimensionality of its domain Θ.

In the domains presented in Sections IV and V, the functions ϕ_i output feature values that encode key aspects of a task—for example distance between the robot body and obstacles in the environment, speed of the motion, or characteristics of a motion planning task. In general, the ϕ_i can either be hand-engineered by a system designer or more generally learned through data-driven approaches [6].

It is important to stress that the misspecification issue we are trying to mitigate is quite general and does not exclusively affect objectives based on hand-crafted features: any model could ultimately fail to capture the underlying motivation of some human actions. While it may certainly be possible, and desirable, to continually increase the complexity of the robot's model to capture a richer space of objectives, there will still be a need to account for the presence of yet-unlearned components of the true objective. In this sense, our work is complementary to open-world objective modeling efforts.

Note that, using a cost model in the form of (10), the observation model (4) becomes overparametrized, since for any (θ, β) pair with $\theta \in \Theta$ and $\beta \in [0, \infty)$, one can always find a different $\theta' = c\theta$ with an associated $\beta' = \beta/c$ leading to the same probability distribution over human choices. This is equivalent to using an unrestricted Θ and $\beta = \|\theta\|$. Due to this overparametrization, the absolute value of β does not have a universal meaning, and restricting θ to have a fixed norm is necessary in order to make comparisons between the β values associated to different θ hypotheses. We thus restrict our Θ to the set of vectors with unit norm.

Consider the case where the human provides input for a cost function in the robot's objective space. This results in the robot inferring high probability on the corresponding θ vector on the unit sphere with a high magnitude β. However, if the cost that the human cares about and provides input for is outside the robot's hypothesis space, the robot will infer low probability on all θ vectors in the unit sphere, with low magnitude βs.

We now proceed by describing the explicit algorithmic approaches to inferring situational confidence in the learning from demonstrations and corrections domains.

IV. Algorithmic Approach: Demonstrations

A. Formulation

In learning from demonstrations, the human directly controls the state trajectory $\mathbf{x}$ through her input $\mathbf{u}_H$, which enables her to offer the robot a demonstration of how to perform the task. Fig. 2 (left) is an example of such a demonstration.

During the demonstration, the robot is often put in gravity compensation mode or is teleoperated, to grant the person full control over the desired trajectory. As such, in this setting, the cost function C_θ does not depend on the robot controls $\mathbf{u}_R$. Additionally, since the person is primarily concerned with the robot's states and not with the (robot or human) actions required to reach those states , we model the human's internal preferences as only dependant on the state trajectory $\mathbf{x}$. Accordingly, the cost function in (10) becomes:

$$C_\theta(\mathbf{x}) = \theta^T \Phi(\mathbf{x}). \tag{11}$$

The cost does not have a direct dependence on the actions, but it has an indirect one, as $\mathbf{x}$ depends on $\mathbf{u}_R$ and $\mathbf{u}_H$.

In our problem formulation, we would like the robot to explicitly reason about how well it can explain the demonstration given its human model. Thus, we can adapt the model in (4) to use this new cost function[2],

$$P(\mathbf{x} \mid \theta, \beta) = \frac{e^{-\beta\theta^T\Phi(\mathbf{x})}}{\int_{\bar{\mathbf{x}}} e^{-\beta\theta^T\Phi(\bar{\mathbf{x}})} d\bar{\mathbf{x}}} \ , \tag{12}$$

then perform the Bayesian update in (5)

$$b'(\theta,\beta) = \frac{P(\mathbf{x} \mid \theta, \beta) b(\theta,\beta)}{\int_{\bar{\theta},\bar{\beta}} P(\mathbf{x} \mid \bar{\theta}, \bar{\beta}) b(\bar{\theta},\bar{\beta}) d\bar{\theta} d\bar{\beta}} \ . \tag{13}$$

Given $b'(\theta,\beta)$, we now can use any of (6), (7) or (8). Next, we discuss making inference with (12) and (13) tractable.

B. Approximation

Although the proposed formalism enables us to capture if the robot's hypothesis space cannot explain the human's input, it is non-trivial to implement tractably for continuous β and θ, and large state and action spaces. Concretely, notice that equations (12) and (13) constitute a doubly-intractable system with denominators that cannot be computed exactly. For this reason, we employ several approximations in order to demonstrate the benefits of estimating situational confidence. Note that we do not consider these a contribution of our work: we choose the simplest approximations that facilitate tractability. There are many methods for approximate inference of θ studied in the literature that could be used for the joint (θ, β) spaces as well, from Metropolis Hastings [16], [39], to acquiring an MLE only via importance sampling of the partition function [6] or via a Laplace approximation [40].

To approximate the intractable integral in (12), we sampled a set $\mathcal{X}$ of 1500 trajectories. We sampled costs according to (11) given by random unit norm θs, then optimized them with an off-the-shelf trajectory optimizer. We used TrajOpt [41], which is based on sequential quadratic programming and uses convex-convex collision checking. This way, we obtain dynamically feasible trajectories that optimize for different features in varying proportions. While this sampling strategy cannot be justified theoretically, it works well in practice: the resulting optimized trajectories are a heuristic for sampling diverse and interesting trajectories in the environment. Future work will address this shortcoming by either providing theoretical guarantees or using importance sampling instead.

For the second approximation to (13), we discretized the space of $\theta \in \Theta$ and $\beta \in \mathcal{B}$ into sets Θ_D and $\mathcal{B}_D$, which leaves us with a finite, easy to compute posterior. For more practical details on specific discretization schemes, see Appendix A-A.

Using the above discretization[3], we can now perform tractable inference from demonstrations $\mathcal{D}$ to obtain a discrete posterior $b(\theta, \beta)$. Algorithm 1 summarizes the full procedure: given $\Theta_D, \mathcal{B}_D, \mathcal{X}$, and $\mathcal{D}$, our method iteratively updates the belief using (12) and (13), resulting in the posterior $b(\theta, \beta)$. Lacking any a-priori information, we chose a uniform prior but our method will work with any prior. We next present examples for what this posterior looks like in different scenarios.

Algorithm 1 Learning from Demonstrations (Offline)

Input: Discretized sets $\Theta_D, \mathcal{B}_D, \mathcal{X}$, set of demonstrations $\mathcal{D}$.
Output: Posterior belief $b(\theta, \beta)$ inferred from $\mathcal{D}$.

```
b(θ, β) ← Uniform(θ, β).
for x in D do
  for all θ ∈ Θ_D, β ∈ B_D do
    P(x | θ, β) = e^{-βθ^T Φ(x)} / Σ_{x̄∈X} e^{-βθ^T Φ(x̄)}  as per (12).
    b(θ, β) ← P(x|θ,β) b(θ,β) / Σ_{θ̄∈Θ, β̄∈B} P(x|θ̄,β̄) b(θ̄,β̄)  as per (13).
  end for
end for
```

C. Examples

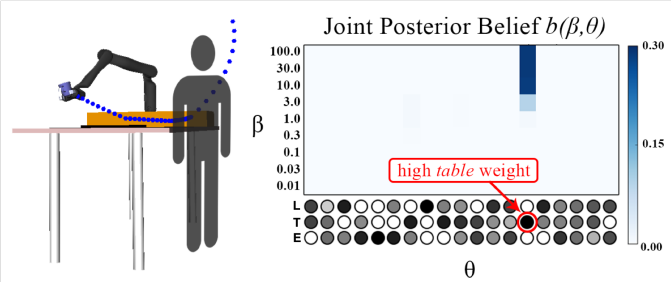
To provide intuition for how situational confidence can indicate when a robot's hypothesis space is misspecified, we illustrate some examples with a robot manipulator learning from a human demonstrator. These examples help prepare the setup we will present in our actual experiments in Section VI.

The robot manipulator is performing a household task of moving cups from a shelf onto the kitchen table. The robot needs to learn from the person's demonstrations how to best perform this task. For this purpose, the person physically guides the robot through one or a few demonstrations of moving the cup down to the table, from which the robot infers the hidden objective function.

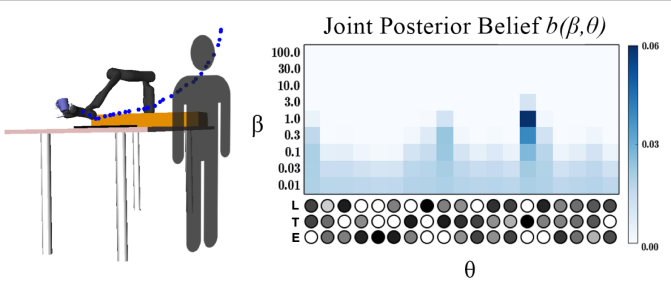
In these examples, the robot's hypothesis space includes three features: efficiency (E) as sum of squared velocities over the trajectory, keeping the cup close to the table (T), and keeping the cup away from the laptop (L) depicted in black.

[2]For deterministic (1), $P(\mathbf{u}_H \mid x^0, \mathbf{u}_R; \theta, \beta)$ is equivalent to $P(\mathbf{x} \mid \theta, \beta)$.

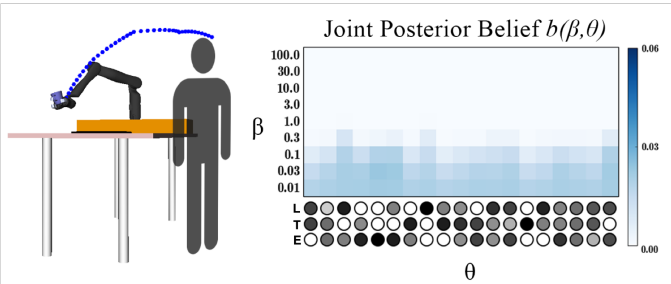
[3]In situations where the designer might want high fidelity inference over a large space of θ vectors, reasoning over a heavily discretized space would be more computationally expensive. However, longer offline computation is possible in our learning-from-demonstrations scenario as the inference happens offline, after providing the robot with human demonstrations. Alternatively, we could use Monte Carlo sampling approaches, similar to [16], [28].



(a) (Left) Simulated perfect demonstration with the objective to keep the cup close to the table. (Right) Posterior belief resulted from this demonstration. Notice that a perfect demonstration leads to a high probability on the correct θ and high values for β.



(b) (Left) Noisy human demonstration with the objective to keep the cup close to the table. (Right) Posterior belief resulted from this demonstration. Notice that a noisy but well-explained demonstration leads to a high probability on the correct θ and moderately high values for β. However, the noise in the demonstration significantly reduces the probability at the distributional peak.



(c) (Left) Simulated perfect demonstration with the objective to keep the cup away from the humans body. (Right) Posterior belief resulted from this demonstration. Notice that, since this demonstration is poorly explained (the robot is not reasoning about distance from the human), the posterior belief is spread out approximately uniformly over all θs and the lowest β values. This indicates that the robot cannot tell what the demonstration was intended for.

Fig. 3: Three examples of demonstrations and the inferred posterior belief after each one of them. The robot infers the right $\theta = [0, 1, 0]$ from the two well-explained demonstrations, but, unlike the perfect simulated demonstration in 3a, the noisy one in 3b cannot reach the highest β and has as overall more spread-out probability distribution with a lower peak value. Lastly, the perfect simulated demonstration that is poorly explained in 3c results in a posterior that is spread-out over all θs and the lowest βs , consistent with the robot not being able to tell what the humans objective was.

Formally, we can represent these three feature mappings as:

$$\Phi(\mathbf{x}) = \begin{bmatrix} \sum_{i=1}^{T}((x^i - x^{i-1})/\Delta t)^2 \\ \sum_{i=0}^{T} ||x^i - x_{\text{table}}||_2 \\ \sum_{i=0}^{T} \max\{0, L - ||x^i - x_{\text{laptop}}||_2\} \end{bmatrix} \tag{14}$$

where L is the radius of a penalty sphere around the laptop, Δt is the discrete timestep between the states in the trajectory, and the corresponding feature weight vector is $\theta \in \mathbb{R}^3$.

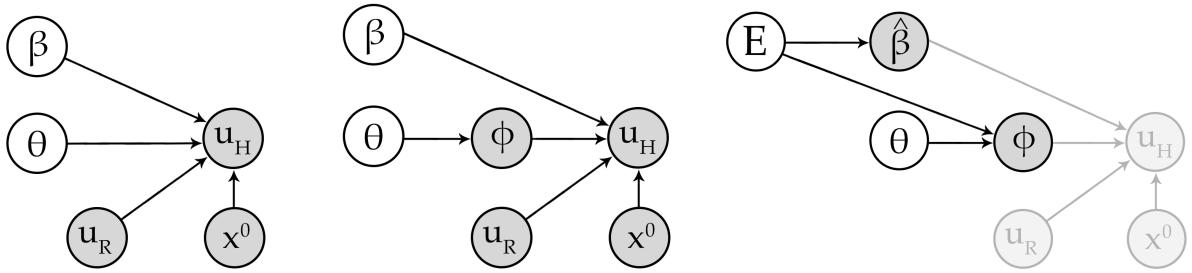
Fig. 3 demonstrates how the feature weight θ and the situational confidence β are affected for well-explained, noisy, and poorly-explained simulated human demonstration. The posterior belief is shown for the combination of discrete parameters θ and β. Higher β values indicate higher situational confidence. The three circles under each column represent the θ vector for that column, with the components being the efficiency, distance from the table, and distance from the laptop features. A larger feature weight is indicated by a darker colored circle, while a white color indicates zero weight.

First, in 3a, we consider the case where the demonstration is a perfectly optimal trajectory produced by TrajOpt [41]. This serves as a sanity check for when the human and the robot have the same hypothesis space and the demonstration is perfect. The optimal demonstration was produced by finding a trajectory that moves the cup from the start configuration to the end while minimizing the distance between the cup and the table. Notice that, with a perfect demonstration, the posterior distribution places the most probability mass on the θ that indicates high penalties for staying away from the table but no penalties for lack of efficiency or closeness to laptop. Moreover, the posterior also reveals that the most likely θ also corresponds *with the highest available confidence* β.

Next, in 3b we recorded a real human demonstration of the same cup-to-table behavior. The nature of demonstrations both on hardware and from real people introduce noise into the demonstration, making it potentially suboptimal with respect to the robot's model. However, in this case the human and the robot still share the same hypothesis space (i.e. the robot and the human both know about the the efficiency, table, and laptop features). Here, we study how the noise in the demonstration affects the robot's inference. Notice that even with an imperfect demonstration, the robot is able to identify the correct θ parameter, but now with a lower confidence β.

Lastly, we consider the example where the demonstration is optimal but the robot does not have a rich enough hypothesis space to explain it. The robot reasons about the same three features, but now the demonstration was produced by optimizing for an additional feature that is outside its hypothesis space: keeping the cup away from the human's body. We observe that the probability distribution in 3c is spread over all the θ values in the space, with the highest values on low βs. This example shows how, in the case of poorly-explained input, the robot's inference is unsure which objective the human had in mind, and assigns low situational confidence to the given input.

These illustrative examples give us valuable insight into how the (θ, β)-belief changes depending on how well-explained the input is. For perfectly explained demonstrations, the inference identifies the correct θ with high posterior probability. As the input becomes more poorly-explained, the robot loses confidence in all θs, assigning approximately uniformly spread-out probability on the lowest situational confidence values β.



(a) In the true graphical model, u_H is an observation of θ and the situational confidence β.

(b) We use the proxy variable Φ to first estimate β efficiently.

(c) We interpret the estimate $\hat{\beta}$ as an indirect observation of the unobserved E, which we then use for the θ estimate.

Fig. 4: Graphical model formulation (a) and modifications to it ((b) and (c)) for real-time tractability.

V. Algorithmic Approach: Corrections

A. Formulation

We consider the setting in which human input is provided in the form of physical interventions during the robot's task execution. Fig. 2 (right) is an example of such a correction. The human may provide a correction to improve some aspect of the task execution that is not represented in the robot's objective space. When the robot receives input, it should be able to reason about its situational confidence in light of the correction and replan its trajectory accordingly for the rest of the task execution or until a new correction happens. Thus, the robot must have access to an inference algorithm that can run in real time. In this section, we will present an online version of our situational confidence framework.

In the physical corrections setting, the robot starts with an initial guess of the parameter θ and uses a trajectory optimization scheme to compute a motion plan seeking to minimize the associated cost C_θ. The robot performs the task at hand by applying controls $\mathbf{u}_R$ via an impedance controller in order to track the computed trajectory $\mathbf{x}$.

At any timestep t during the trajectory execution, the human may physically interact with the robot, inducing a joint torque u_H^t. When this happens, the robot can use the human input to update its estimated θ parameter, and thereby the corresponding objective C_θ. Given the new adapted objective, the robot replans an optimized trajectory $\mathbf{x}$ and tracks it until the next human input is sensed or until the task is completed.

Following [13], the robot's representation of the task assumes that the human does not explicitly care about the robot's control effort, but only about features of the state trajectory. In addition, the human is assumed to have a preference for minimizing her own control effort. This captures the human's incentive to have the robot perform the task autonomously, providing only minimal input to guide the robot towards the correct behavior when necessary. Encompassing these assumptions, the cost (10) takes the form:

$$C_\theta(\mathbf{x}, u_H^t) = \theta^T \Phi(\mathbf{x}) + \lambda \|u_H^t\|^2. \tag{15}$$

To approximately compute the trajectory resulting from the human's input, we follow the approach in [13] and introduce the notion of a *deformed trajectory* $\mathbf{x}_D$. This trajectory constitutes the robot's estimate of the human's desired trajectory given her applied torque u_H^t. Given the robot's default trajectory $\mathbf{x}_R := \mathbf{x}(\cdot; x^0, \mathbf{u}_R, \mathbf{0})$ and having observed the instantaneous human intervention u_H^t, we compute $\mathbf{x}_D$ by deforming the robot's default trajectory in the direction of u_H^t:

$$\mathbf{x}_D = \mathbf{x}_R + \mu A^{-1} \tilde{\mathbf{u}}_H \ , \tag{16}$$

where $\mu > 0$ scales the magnitude of the deformation, $A \in \mathbb{R}^{n(T+1)\times n(T+1)}$ defines a norm on the Hilbert space of trajectories[4] and dictates the deformation shape [42], and $\tilde{\mathbf{u}}_H \in \mathbb{R}^{n(T+1)}$ is u_H^t at indices nt through $n(t+1)$ and 0 otherwise. The human is therefore modeled by (15) as trading off between inducing a good trajectory $\mathbf{x}_D$ with respect to θ, and minimizing her effort.

Equipped with this cost function, we need the robot to reason about the reliability of its objective space given new inputs in the form of corrections. In contrast with our analysis in Section IV, here the person does not give full demonstrations $\mathbf{x}$, but instead offers corrections u_H^t based on the robot's default trajectory $\mathbf{x}_R$. Applying (4) to this setting, we have:

$$P(u_H^t \mid x^0, \mathbf{u}_R; \theta, \beta) = \frac{e^{-\beta(\theta^\top \Phi(\mathbf{x}_D) + \lambda \|u_H^t\|^2)}}{\int e^{-\beta(\theta^\top \Phi(\bar{\mathbf{x}}_D) + \lambda \|\bar{u}\|^2)} d\bar{u}} \ , \tag{17}$$

where $\mathbf{x}_D$ and $\bar{\mathbf{x}}_D$ are given by (16) applied to their respective controls u_H^t and $\bar{u}$.

Ideally, with this model of human actions, illustrated in Fig. 4a, we would perform inference over both the situational confidence β and the modeled parameters θ by maintaining a joint Bayesian belief $b'(\theta, \beta)$. Analogously to the demonstrations case, our probability distribution over θ would automatically adjust for well-explained corrections, whereas for poorly-explained ones the robot's posterior would not deviate significantly form its prior on θ. Unfortunately, this Bayesian update is not generally feasible in real time, given the continuous and possibly high-dimensional nature of the parameter space Θ. Even in simple scenarios with a small number of continuous features, discretizing Θ as we did in the demonstrations case would generally yield an overly slow inference, making the method impractical for use in the real-time collaborative scenarios that we are interested in here.

[4] We used a norm A based on acceleration, consistent with [13], but other norm choices are possible as well.

Thus, to evaluate the benefits of estimating β we need to derive an online method that goes beyond simple discretization.

B. Approximation

To alleviate the computational challenge of performing joint inference over β and θ, we introduce a structural assumption that will enable us to approximately decouple the two inference problems.

1) Estimating β: To estimate β without dependence on θ, we will assume that in order to decide what correction to provide, the human will first choose the desired features Φ of the resulting trajectory $\mathbf{x}_D$ and then select an input u_H^t that will obtain these features (Fig. 4b).

Based on the observed human input u_H^t and the trajectory features of the deformed trajectory $\Phi(\mathbf{x}_D)$, the robot can obtain an estimate of β by considering how efficient the human's input was for the features achieved. Letting $\mathcal{U}_\Phi$ be the set of inputs that achieve the same observed features $\Phi_D := \Phi(\mathbf{x}_D)$, the Boltzmann decision model gives

$$P(u_H^t \mid x^0, \mathbf{u}_R; \Phi_D, \beta) = \frac{e^{-\beta(\theta^\top \Phi_D + \lambda \|u_H^t\|^2)}}{\int_{\mathcal{U}_\Phi} e^{-\beta(\theta^\top \Phi(\bar{\mathbf{x}}_D) + \lambda \|\bar{u}\|^2)} d\bar{u}} = \frac{e^{-\beta\lambda\|u_H^t\|^2}}{\int_{\mathcal{U}_\Phi} e^{-\beta\lambda\|\bar{u}\|^2} d\bar{u}} \ , \tag{18}$$

since the term $\theta^\top \Phi(\bar{\mathbf{x}}_D)$ is constant for all $\bar{u} \in \mathcal{U}_\Phi$ and equal to the term $\theta^\top \Phi_D$ in the numerator.

Using (18), the robot can obtain an estimate of β by considering how efficient the human's correction was for the features achieved—if the input seems highly inefficient, this is indicative that the features modeled by the robot may not accurately capture the human's preference.

It is useful to approximate the integral over the constrained set $\mathcal{U}_\Phi \subset \mathcal{U}$ by an integral over the entire set of possible inputs $\mathcal{U}$, introducing a penalty term in the exponent that results in a soft indicator function for $\bar{u} \in \mathcal{U}_\Phi$:

$$P(u_H^t \mid x^0, \mathbf{u}_R; \Phi_D, \beta) \approx \frac{e^{-\beta\lambda\|u_H^t\|^2}}{\int_{\mathcal{U}} e^{-\beta(\lambda\|\bar{u}\|^2 + \kappa\|\Phi(\bar{\mathbf{x}}_D) - \Phi_D\|^2)} d\bar{u}} \ . \tag{19}$$

Note that for an arbitrarily large κ there is an arbitrarily small probability assigned to $\mathcal{U} \backslash \mathcal{U}_\Phi$ in the integral. It is now possible to apply the Laplace approximation to the unconstrained integral (see Appendix B for details), yielding:

$$P(u_H^t \mid x^0, \mathbf{u}_R; \Phi_D, \beta) \approx \frac{e^{-\beta\lambda\|u_H^t\|^2}}{e^{-\beta(\lambda\|u_H^*\|^2 + \kappa\|\Phi(\mathbf{x}_D^*) - \Phi_D\|^2)}} \sqrt{\frac{\beta^k |H_{u_H^*}|}{2\pi^k}} \ , \tag{20}$$

where k is the action space dimensionality and $H_{u_H^*}$ is the Hessian of the exponent in the denominator of (19) around u_H^*. We obtain the optimal action u_H^* by solving the constrained optimization problem (see Appendix A-B):

$$\begin{aligned} \underset{\tilde{u}_H}{\text{minimize}} \quad & \|\tilde{u}_H\|^2 \\ \text{subject to} \quad & \Phi(\mathbf{x} + \mu A^{-1} \tilde{\mathbf{u}}_H) - \Phi_D = 0 \ . \end{aligned} \tag{21}$$

In other words, the resulting u_H^* is the minimal norm $\tilde{u}_H$ the human could have taken, constrained to lie in $\mathcal{U}_\Phi$. As such, the second norm in the denominator's exponent is 0, and the final conditional probability becomes:

$$P(u_H^t \mid x^0, \mathbf{u}_R; \Phi_D, \beta) = e^{-\beta\lambda(\|u_H^t\|^2 - \|u_H^*\|^2)} \sqrt{\frac{\beta^k |H_{u_H^*}|}{2\pi^k}} \ . \tag{22}$$

We derive below the maximum likelihood estimator (MLE), noting that a maximum *a posteriori* (MAP) estimator is often appropriate given a certain prior on β.

$$\begin{aligned} \hat{\beta} &= \arg\max_\beta \{\log(P(u_H^t \mid x^0, \mathbf{u}_R; \Phi_D, \beta)\} \\ &= \arg\max_\beta \{-\beta\lambda(\|u_H^t\|^2 - \|u_H^*\|^2) + \log(\sqrt{\frac{\beta^k |H_{u_H^*}|}{2\pi^k}})\}. \end{aligned} \tag{23}$$

Applying the first-order condition and setting the derivative to zero yields the maximizer:

$$\hat{\beta} = \frac{k}{2\lambda(\|u_H^t\|^2 - \|u_H^*\|^2)} \ . \tag{24}$$

The estimator[5] above yields a high value when the difference between u_H^t and u_H^* is small, i.e. the person's correction achieves the induced features $\Phi(\mathbf{x}_D)$ efficiently. For instance, if $\mathbf{x}_D$ brings the robot closer to the table, and u_H^t pushes the robot straight towards the table, u_H^t is an efficient way to induce those new feature values. However, when there is a much more efficient alternative (e.g. when the person pushes mostly sideways rather than straight towards the table), $\hat{\beta}$ will be small. Efficient ways to induce the feature values will suggest well-explained inputs, inefficient ones will suggest poorly-explained corrections.

2) Estimating θ: To tractably estimate θ building on the β estimate, we introduce an auxiliary binary variable $E \in \{0, 1\}$ indicating whether the human's intervention can be well *explained* by the robot's modeled cost features. We will perform offline training with ground-truth access to this variable in order to learn its relation to the robot's estimate $\hat{\beta}$.

When $E = 1$, the human's desired modification of the robot's behavior can be well explained by *some* vector $\theta \in \Theta$, which will lead the intervention to appear less noisy to the robot (i.e. β is large). As a result, the correction u_H^t is likely to be efficient for the cost encoded by this θ. Conversely, when $E = 0$, the intervention appears noisy (i.e. β is small), and the human's correction cannot be well explained by any of the cost features modeled by the robot.

The graphical model depicted in Fig. 4c relates the induced feature values Φ_D to θ as a function of the E. When $E = 1$, the induced features will tend to have low cost with respect to θ; when $E = 0$, the induced features *do not depend on* θ, and we model them as Gaussian noise centered around the feature values of the robot's currently planned trajectory $\mathbf{x}_R$.

$$P(\Phi_D \mid \theta, E) = \begin{cases} \dfrac{e^{-\theta^\top \Phi_D}}{\int e^{-\theta^\top \Phi(\tilde{\mathbf{x}}_D)} d\tilde{\mathbf{x}}_D}, & E = 1 \\ \left(\frac{\nu}{\pi}\right)^{\frac{k}{2}} e^{-\nu||\Phi_D - \Phi(\mathbf{x}_R)||^2}, & E = 0 \end{cases} \tag{25}$$

[5]Note that $\hat{\beta}$ is non-negative, since u_H^* is the minimal-norm $\tilde{u}_H$ that satisfies the constraint, so the difference in the denominator of (24) is positive.

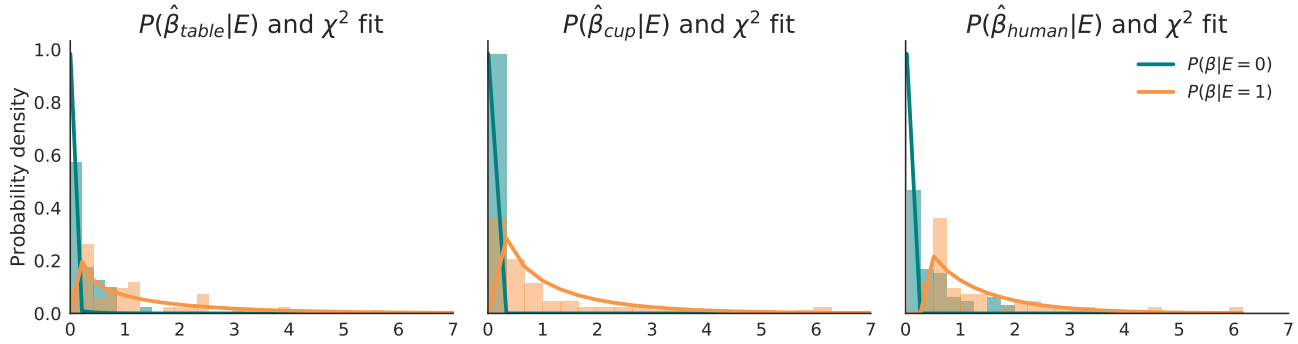


Fig. 5: Empirical estimates for $P(\hat{\beta} \mid E)$ and their corresponding chi-squared (χ^2) fits.

with the constant in the $E = 0$ case corresponding to the normalization term of the normal distribution.

In addition, this graphical model relates the $\hat{\beta}$ resulting from the model in Fig. 4b to E by a $P(\hat{\beta} \mid E)$. We fit this distribution from controlled user interaction samples where we have ground-truth knowledge of E[6]. For each sample interaction, we compute $\hat{\beta}$ (for example, using (24) if using MLE) and label it with the corresponding binary E value. We fit a chi-squared distribution to these samples to obtain the probability distributions for $P(\hat{\beta} \mid E = 0)$ and $P(\hat{\beta} \mid E = 1)$. The resulting distributions are shown in Fig. 5[7].

Using the model in Fig. 4c with the learned distribution $P(\hat{\beta} \mid E)$, we can infer a θ estimate in real time whenever a physical correction from the human is measured. We do this tractably by interpreting the estimate $\hat{\beta}$ obtained from (24) as an indirect observation of the unknown variable E. We combine the empirically characterized likelihood model $P(\hat{\beta} \mid E)$ with an initial uniform prior $P(E)$ to maintain a Bayesian posterior on E based on the evidence $\hat{\beta}$ constructed from human observations at deployment time, $P(E \mid \hat{\beta}) \propto P(\hat{\beta} \mid E)P(E)$.

Further, since we wish to obtain a posterior estimate of the human's objective θ, we use the model from Fig. 4c to obtain the posterior probability measure

$$P(\theta \mid \Phi_D, \hat{\beta}) \propto \sum_{E \in \{0,1\}} P\big(\Phi_D \mid \theta, E\big) P(E \mid \hat{\beta}) P(\theta) \ . \quad (26)$$

Following [13], we note that we can approximate the partition function in the human's policy (25) by employing the Laplace approximation. Taking a second-order Taylor series expansion of the exponent's objective about $\mathbf{x}_R$, the robot's current best guess at the optimal trajectory, we obtain a Gaussian integral that can be evaluated in closed form

$$P(\Phi_D \mid \theta, E = 1) \approx e^{-\theta^\top \left(\Phi_D - \Phi(\mathbf{x}_R)\right)} \ . \quad (27)$$

We also consider a Gaussian prior distribution of θ around the robot's current estimate $\hat{\theta}$:

$$P(\theta) = \frac{1}{(2\pi\alpha)^{\frac{k}{2}}} e^{-\frac{1}{2\alpha}||\theta - \hat{\theta}||^2} \ , \quad (28)$$

where $\alpha \geq 0$ determines the variance of the Gaussian.

To obtain an update rule for the θ parameter, we can simply plug (25), (27), and (28) into (26). For legibility, let's denote $\Gamma(\Phi_D, E = i) = P(E = i \mid \hat{\beta}) P\big(\Phi_D \mid \theta, E = i\big)$, for $i \in \{0, 1\}$. Then, the maximum-a-posteriori estimate of the human's objective θ is the solution maximizer of

$$\begin{aligned} &P(\theta)\Big[\Gamma(\Phi_D, E = 1) + \Gamma(\Phi_D, E = 0)\Big] \\ &\quad = \frac{1}{(2\pi\alpha)^{\frac{k}{2}}} e^{-\frac{1}{2\alpha}||\theta - \hat{\theta}||^2} \Big[P(E = 1 \mid \hat{\beta}) e^{-\theta^\top \left(\Phi_D - \Phi(\mathbf{x}_R)\right)} \\ &\qquad + P(E = 0 \mid \hat{\beta}) \left(\frac{\nu}{\pi}\right)^{\frac{k}{2}} e^{-\nu||\Phi_D - \Phi(\mathbf{x}_R)||^2}\Big] \ . \end{aligned} \quad (29)$$

Differentiating (29) with respect to θ and equating to 0 gives the maximum-a-posteriori update rule

$$\hat{\theta}' = \hat{\theta} - \alpha \frac{\Gamma(\Phi_D, E = 1)}{\Gamma(\Phi_D, E = 1) + \Gamma(\Phi_D, E = 0)} \big(\Phi_D - \Phi(\mathbf{x}_R)\big) \ . \quad (30)$$

We note that due to the coupling in $\hat{\theta}'$, the solution to (30) is non-analytic and can instead be obtained via numerical approaches like Newton-Raphson or quasi-Newton methods.

In previous objective-learning approaches including [13] and [37], it is implicitly assumed that all human actions are fully explainable by the robot's representation of the objective function space ($E = 1$), leading to the simplified update

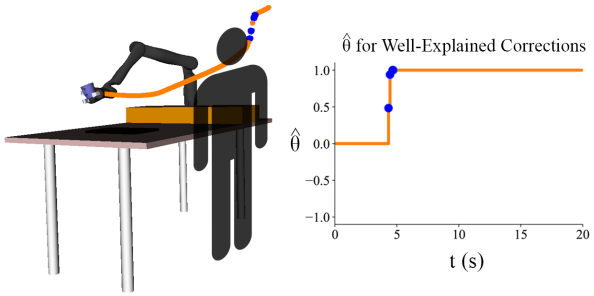
$$\hat{\theta}' = \hat{\theta} - \alpha\big(\Phi_D - \Phi(\mathbf{x}_R)\big) \ , \quad (31)$$

which can be easily seen to be a special case of (30) when $P(E = 0 \mid \hat{\beta}) \equiv 0$. Our proposed update rule therefore *generalizes* commonly-used objective-learning formulations to cases where the human's underlying objective function is not fully captured by the robot's model. We expect that this extended formulation will enable learning that is more robust to misspecified or incomplete human objective parameterizations.[8] Once we obtain the $\hat{\theta}'$ update, we replan the robot trajectory in its 7-DOF configuration space with an off-the-shelf trajectory optimizer, TrajOpt [41].

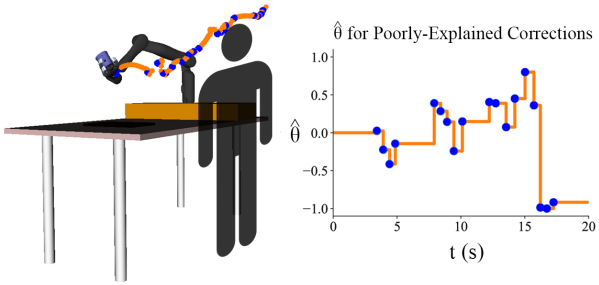
[6]Since we tell users what to optimize for, we know whether the human's input is well-explained with respect to the robot's hypothesis space or not.

[7]Because users tend to accidentally correct more than one feature, we perform β-inference separately for each feature. This requires more overall computation (although still linear in the number of features, and can be parallelized) and a separate $P(\hat{\beta} \mid E)$ estimate for each feature.

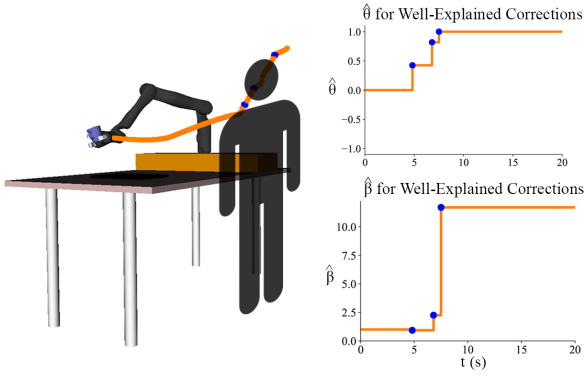
[8]Note that to enforce the constraint on $||\theta|| = 1$, we can indeed project the resulting $\hat{\theta}'$ onto the unit ball. In practice, because our learning from corrections algorithm separates the β-inference from the θ-inference, this projection is no longer required, but we found it helpful to still constrain the space of Θ to encourage smoothness in the change of the cost function.



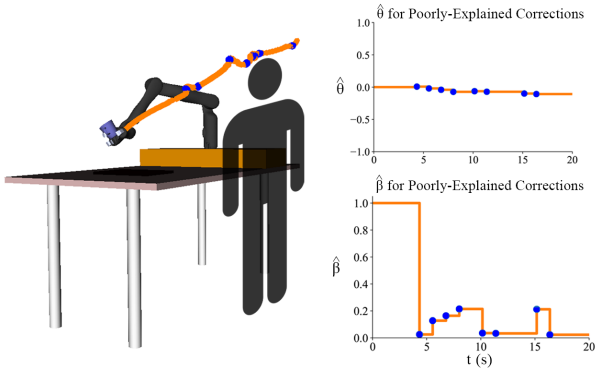
(a) (Left) The human applies well-explained corrections to keep the cup close to the table. Learning with fixed β leads to a correct trajectory that satisfies the human's preference. (Right) As the person corrects the robot by pushing down on it, the learning algorithm gradually updates its weight on the feature modeling distance to table.



(b) (Left) The human applies poorly-explained corrections to keep the cup upright. Learning with fixed β leads to a oscillatory and noisy trajectory. (Right) Here the learning algorithm incorrectly updates the weight on the distance to table feature, leading to unintended learning.



(c) (Left) The human applies well-explained corrections to keep the cup close to the table. Learning with estimated β leads to a correct trajectory that satisfies the human's preference. (Right) As the person corrects the robot by pushing down on it, the learning algorithm infers high $\hat{\beta}$ and gradually updates its weight on the feature modeling distance to table.



(d) (Left) The human applies poorly-explained corrections to keep the cup upright. Learning with estimated β leads to a smooth trajectory where the robot is robust to poorly-explained inputs. (Right) Here the learning algorithm infers low $\hat{\beta}$ and correctly avoids unintended learning for the distance to table feature.

Fig. 6: Examples of physical corrections (interaction points shown in blue) and the resulting behavior for the fixed β method (top) and estimated β method (bottom). When the corrections are well explained, both methods learn the correct weight $\hat{\theta} = 1.0$. In the case of poorly-explained corrections, our method infers low $\hat{\beta}$ and manages to reduce unintended learning, whereas the fixed β method produces incorrect oscillatory behavior.

The update rule changes the weights in the objective in the direction of the feature difference as well, but how much it does so depends on the probability assigned to the correction being well-explained. Looking back at Section III, this update is approximating (7). At one extreme, if we know with full certainty that the correction is well explained, then we do the full update as in traditional objective learning. But crucially, at the other extreme, if we know that the correction is poorly explained, we do not update at all and keep our prior belief.

Overall, the full algorithm is given in Algorithm 2. The robot begins tracking a trajectory $\mathbf{x}_R$ given by an initial $\hat{\theta}$. Once a human torque u_H is sensed, the robot deforms its trajectory to compute the induced features Φ_D, computes the optimal human action u_H^* using (21), and uses it to estimate $\hat{\beta}$ for that input. It then updates $\hat{\theta}$ using the learned distributions $P(\hat{\beta} \mid E = i), \forall i \in \{0, 1\}$, and updates its tracked trajectory $\mathbf{x}_R$. For more practical details on how replanning works, and how to set various hyperparameters, consult Appendix A-B.

Algorithm 2 Learning from Corrections (Online)

Input: $P(\hat{\beta} \mid E = i), \forall i \in \{0, 1\}$ from training data.
  Initialize $\mathbf{x}_R \leftarrow TrajOpt(\hat{\theta})$ for initial $\hat{\theta}$.
  while goal not reached **do**
    if $u_H \neq \mathbf{0}$ **then**
      $\mathbf{x}_D = \mathbf{x}_R + \mu A^{-1} \tilde{\mathbf{u}}_H$.
      $u_H^* \leftarrow OptimalHumanAction(\Phi_D)$, as per (21) .
      $\hat{\beta} = \frac{k}{2\lambda(\|u_H\|^2 - \|u_H^*\|^2)}$.
      $\hat{\theta} \leftarrow \hat{\theta} - \alpha \frac{\Gamma(\Phi_D, E=1)}{\Gamma(\Phi_D, E=1) + \Gamma(\Phi_D, E=0)} \big(\Phi_D - \Phi(\mathbf{x}_R)\big)$.
      $\mathbf{x}_R \leftarrow TrajOpt(\hat{\theta})$.
    end if
  end for

C. Examples

As in Section IV, we now illustrate some examples to help lay out some of the setup we will present in our actual experiments in Sections VI and VII. We provide intuition for

how the estimators of β and θ work when we have a perfectly specified objective space and a misspecified objective space. In all of the examples, the robot reasons about the previously described distance from the table feature. What changes is the feature for which the human decides to provide corrections.

We look at two situations: the human may correct the relevant feature and push the robot closer to the table, or she might provide an poorly-explained input to keep the coffee mug upright. Fig. 6 illustrates the two scenarios and contrasts our estimated-β approach to the state of the art fixed-β approach that uses (31).

On the top we present the fixed-β method and its performance with both the well-explained and the poorly-explained input. When the input is well explained, the left image shows that the robot learns from the interactions and converges close to the true $\theta = 1$. However, when the input is poorly explained on the right, the robot incorrectly learns fictitious θ values and produces oscillatory behavior.

In the bottom row of Fig. 6 we present our described estimated-β method. In the case of well-explained inputs, the value for $\hat{\beta}$ increases, allowing θ to grow up to the real value $\theta = 1$. The method has the same behavior as the state of the art. However, more importantly, in the case of poorly-explained input, our method immediately estimates low $\hat{\beta}$ values, which allows it to significantly reduce unintended learning as compared to the state of the art.

This figure illustrates how situational confidence estimation can aid the robot when the human input is poorly explained. We stress that although our method does not allow the robot to magically learn the correct behavior that the user desires, it greatly reduces unintended learning and undesired behaviors.

VI. Case Studies

Equipped with our algorithmic approaches to situational confidence estimation, we now consider two case studies in learning from demonstrations and corrections using real human input on a 7-DoF robot manipulator.

A. Demonstrations

We collected human demonstrations of household motion planning tasks and performed our situational confidence inference offline. We recruited 12 people to physically interact with a JACO 7-DoF robotic arm and analyzed 4 common cases that can arise in the context of personal robotics learning.

For all the experiments in this section, we asked the participants to provide demonstrations with respect to a feature of interest, which the robot might (well-explained) or might not (poorly-explained) have in its hypothesis space. Some of the features that the humans had to prioritize include: distance of the end effector from the table, distance from the person, or distance from the end-effector to a laptop placed on the table.

Before giving any demonstrations, each person was allowed a period of training with the robot in gravity compensation mode to get accustomed to interacting with the robot. When collecting human demonstrations, participants were asked to move the robot arm holding a cup of coffee from the upper shelf of a cupboard to right above the table, across a laptop.

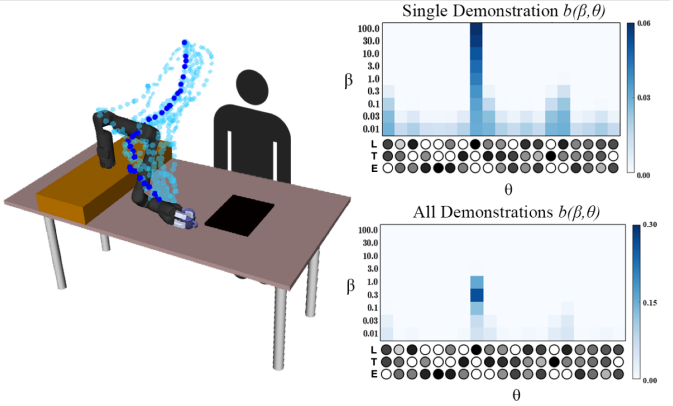


Fig. 7: (Left) Human demonstrations avoiding the laptop. (Right) Upper distribution is the posterior belief for the highlighted blue demonstration. Since the robot has the laptop feature in its hypothesis space, this demonstration induces a high β on the correct $\theta = [0, 0, 1]$. Below, when considering all the demonstrations, the inference procedure converges to a slightly lower β value due to the suboptimality of some of the demonstrations in the dataset.

After collecting all demonstrations, we designed the robot's hypothesis space for inference purposes. In all four scenarios that we will illustrate, the robot reasons over the same three features as in (14): E, T, and L. Although the robot always knows about these features, the demonstrations may have been given relative to different (and potentially unmodeled) features.

Throughout our scenarios, we tested two hypotheses:

H1. *If the human input is well-explained, our inference procedure places high probability on the correct θ hypothesis, with a high situational confidence β.*

H2. *If the human input is poorly-explained, our inference procedure does not place high probability on any θ hypothesis and is uniform over all hypotheses with low situational confidence β.*

To test these hypotheses, we looked at the resulting inferred belief. Given the demonstrations and a parametrization of the cost function, we first updated the belief over the weight and situational confidence parameters for each single demonstration, $b_{single}(\theta, \beta)$. This gives insights into how a single demonstration can affect the robot's inference procedure.

Next, we used all 12 human demonstrations to obtain a probability distribution over the weight and confidence measures, $b_{all}(\theta, \beta)$ for each scenario. By using multiple demonstrations as evidence about the cost and the situational confidence parameter, we see how in some scenarios multiple demonstrations can help improve confidence in the θ estimation.

We now present experimental results in two scenarios that support our above hypotheses.

1) Well-specified objective space: Here we consider a scenario where the robot and the human share the same hypothesis space, i.e. the robot's model is well specified. The participants were instructed to avoid spilling the coffee over the laptop by providing a demonstration where the robot's end-effector is away from the electronic device. Here, the feature of interest was the distance from the laptop which was in the robot's

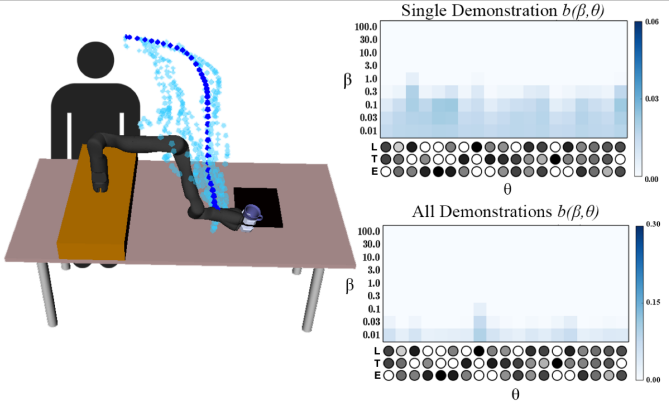


Fig. 8: (Left) Human demonstrations avoiding the user's body. (Right) Upper distribution is the posterior belief $b(\beta, \theta)$ for the highlighted demonstration. Since the robot's model does not include distance to the user's body, none of the robot's hypotheses can explain the demonstration, as reflected in the higher probabilities on low βs for all θs. After performing inference on all the demonstrations, the distribution in the lower right plot shows even more probability mass on the lowest situational confidence values.

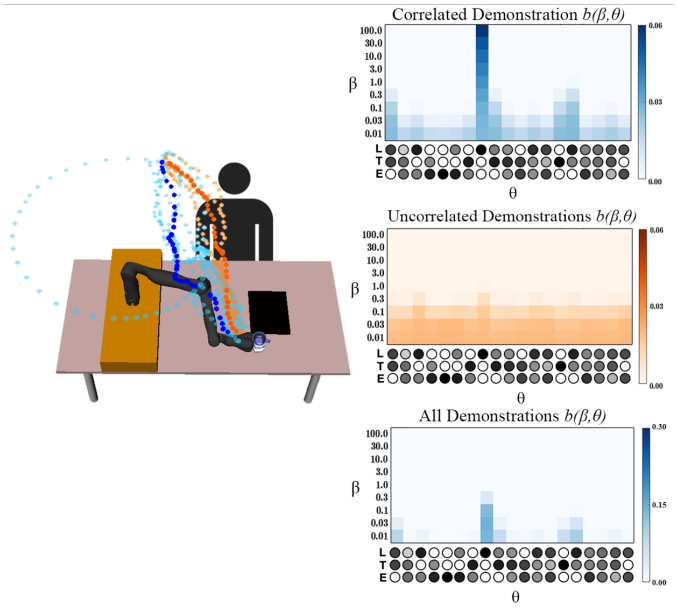


Fig. 9: (Left) Human demonstrations avoiding the user's body. The blue cluster is correlated with the feature describing distance from the laptop. The orange cluster is uncorrelated. (Right) The top distribution is the posterior belief $b(\beta, \theta)$ for the highlighted blue correlated demonstration. Notice that the hypothesis that puts all weight on avoiding the laptop $\theta = [0, 0, 1]$ dominates the distribution. Meanwhile, the posterior belief for the highlighted orange demonstration indicates low situational confidence in all hypotheses. The bottom distribution shows that when combining all demonstrations, the robot continues to have low situational confidence although the laptop hypothesis has slightly higher β.

hypothesis space: the demonstration would be well explained as long as the demonstration maintained a distance of at least L meters away from the center of the laptop.

On the left of Fig. 7 we visualize all 12 recorded demonstrations and the experimental setup. Note that most participants had an easy time providing demonstrations which avoided the laptop. Indeed, we noticed that 10 out of the 12 demonstrations resulted in high situational confidence and a probability distribution similar to the one at the top right of Fig. 7. Here, the θ vector that has largest weight on the third feature (distance from the laptop) is correctly inferred to have high β value. This signals that the robot is highly confident the person provided a demonstration that avoids the laptop, which is correct and supports our hypothesis H1.

Another interesting observation is that the situational confidence over all 12 demonstrations together is lower than in the case of the single optimal demonstration highlighted in blue (peak at around 1.0 instead of 100.0)[9]. This is due to the two noisy demonstrations that came too close to the laptop. When working with non-expert users, it is inevitable that such imperfect demonstrations will arise. However, despite the challenge of noisy and/or erroneous demonstrations, the algorithm recovers the correct θ hypothesis with a relatively high β, supporting H1 once again.

2) Misspecified hypothesis space: We look at the opposite scenario: the robot and the human do not share the same hypothesis space and the robot's model is clearly misspecified.

Participants were instructed to move the robot from start to end while also keeping the robot's hand away from their body to avoid spilling coffee on their clothes. Since the robot's cost function does not include any notion of distance to humans, the demonstrations should appear poorly explained relative to the robot's model of how humans choose demonstrations.

Fig. 8 visualizes all 12 demonstrations as well as the posterior probability distributions for a single highlighted trajectory and for all 12. For both a single demonstration and all of them, in the case of misspecification none of the hypotheses are correct. Thus, the robot infers equally low probability for all θs, with low situational confidence, supporting our hypothesis H2. This signals that the robot is unsure what the person's demonstration referred to, as we expected.

These two examples illustrate cases where our method supports the two hypotheses above. However, there are important limitations that we discuss in the following two scenarios.

3) Feature correlation: The past two examples demonstrate clear instances when the robot's objective space is either well specified or misspecified. However, often times situations will be more ambiguous. For example, although the human input may refer to a feature that is nonexistent in the robot's hypothesis space, the robot may know about a feature that is correlated to the one the human is trying to affect. In this next scenario, we investigate how such feature correlation influences the situational confidence estimates.

We asked the participants to move the robot from the same

[9]In the lower right belief in Fig. 7, note from the colorbar values that the probability mass is more peaked than in the case of a single demonstration. This confirms our intuition that the robot's certainty in the hypothesis is enhanced the more demonstrations supporting that hypothesis it receives.

start and end as before, while keeping the cup in the robot's end-effector away from their body to avoid spilling coffee on their clothes. The setup is similar to the poorly-explained demonstration in the previous scenario, only that now the human starts in a different initial position.

Visualizations of the 12 demonstrations in Fig. 9 showcase that although all demonstrations move the cup away from the person, some of them (depicted in blue) also maintain a good distance away from the laptop. Hence, even though the human was trying to teach the robot to stay away from their body, the robot interprets the human's demonstrations as a signal to stay away from the laptop. Thus, we say that the distance from human and distance from laptop features are *correlated*.

When looking at the top-right posterior probability in Fig. 9, the distribution over θ, β shows that our algorithm infers a high situational confidence for the θ that fully considers the distance from the laptop. Thus, even if the human input does not pertain to a feature in the robot's hypothesis space, in some cases the demonstration can still be explained via correlated features in the robot's hypothesis space. This observation does not support H2 and is clearly a limitation of our method.

However, the orange cluster of demonstrations in Fig. 9, showcase the fine line between demonstrations that induce a feature correlation and those that do not. The orange demonstrations clearly ignore the laptop and simply take the shortest path to the end goal while avoiding the human's body. As we can see in the orange probability distribution, our method infers a uniform distribution over all θ hypotheses, with a focus on the lowest situational confidence values, backing H2.

These two clusters highlight that our method infers reasonable θ, β values even in the case of feature correlation. The robot either infers a good θ to perform its original task through the means of another feature, or it has low confidence in understanding the input.

When we look at the posterior distribution that results from all 12 demonstrations, the bottom-right part of the figure shows that, due to the correlation in the blue cluster, there is increased probability on the θ that considers fully the distance from the laptop. However, due to the ambiguity of the orange cluster, the situational confidence is not as high as it would be in a well-explained case (see Fig. 7).

4) Feature engineering: Many of the cost function features we considered so far have been intuitive to provide demonstrations for. However, some cost functions may be particularly challenging or unintuitive for human users. Two extreme examples of this could be features learned using complex function approximators or unintuitive features like minimizing the total energy of a system.

In our scenario, the feature users have a difficult time providing good demonstrations for is the distance between the robot's hand and the table along the trajectory. Since the feature was designed as the sum of distances to table for all waypoints in the trajectory, the optimal demonstration immediately moves the end-effector to the table and then keeps it right above the tabletop for the rest of the path, as seen in Fig. 3a. This limitation does not support H1.

However, this mathematical optimum does not necessarily align with how human users interpret the best behavior for this task. In our experiments, most users gradually bring the robot's hand closer to the table, rather than pushing it down immediately, for a more smooth and natural motion (see left in Fig. 10). These demonstrations thus appear noisy and suboptimal with respect to the robot's model and make it difficult to infer the true θ from a single demonstration.

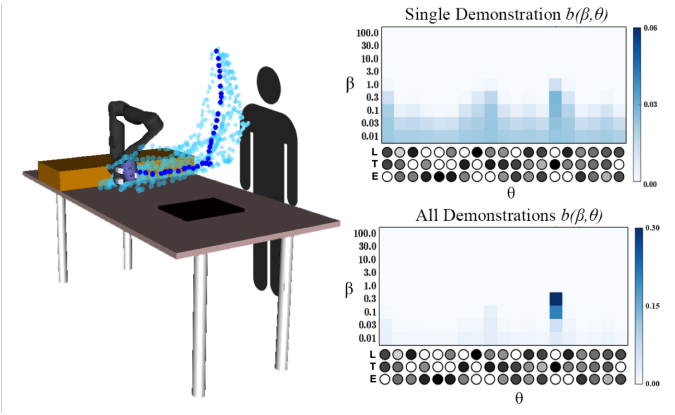


Fig. 10: (Left) Human demonstrations keeping the cup in the end-effector close to the table. (Right) Because it is difficult for the person to give a good demonstration, the top posterior does not have a clearly defined peak for one particular hypothesis, although several θs are favored. In the bottom distribution, we notice that when presented with all 12 demonstrations, the robot can more clearly infer the correct hypothesis for the distance to the table, $\theta = [0, 1, 0]$.

This phenomenon is reflected more clearly when we look at the top-right belief distribution in Fig. 10. Although the distribution for the highlighted blue demonstration has some peaks around hypotheses that strongly favor the feature responsible for distance to the table, it is not nearly as clearly defined as it should be for a well-explained demonstration (see Fig. 7).

However, when the robot gathers evidence from multiple demonstrations, the algorithm does manage to figure out that this is the feature that people were optimizing for. The bottom right plot in Fig. 10 illustrates that, once again, having more input samples eventually leads our algorithm to converge to a strong probability for the right θ with a reasonably high β. Although our method cannot back H1 when inferring the objective from a single demonstration, more data leads our algorithm to correctly support H1.

Summary: The four situations presented above illustrate that our two original hypotheses H1 and H2 are supported most of the time (VI-A1, VI-A2), with some exceptions (VI-A3, VI-A4). We saw that when the person has a difficult time giving a good demonstration (Section VI-A4), our method cannot support H1 unless provided with multiple demonstrations, to disambiguate the inherent noise in the user's suboptimal input. Additionally, when the person provides what should be a poorly-explained demonstration (Section VI-A3), feature correlation might lead the inference to falsely detect θs corresponding to that input, contradicting H2. However, we observed that when given more demonstrations, our algorithm can attribute low situational confidence β if the uncorrelated

input is sufficient. More work is needed in this area.

B. Corrections

We now turn our attention to case where human input is sparse and in the form of intermediate corrections during the robot's task execution. Here we present an offline case study where we analyze how our estimates of $\hat{\beta}$ enable us to distinguish if the input is well explained or not to the robot's model of the human. For a full exploration of the real-time updates from human corrections, we conduct an online user study which we later describe in Section VII.

We recruited 12 additional individuals to physically interact with the same robotic manipulator. Each participant was asked to intentionally correct a feature (that the robot may or may not have in its hypothesis space): adjusting the distance of the end effector from the table, adjusting the distance from the person, or adjusting the cup's orientation.

During this case study the robot did not attempt to update the feature weights θ and simply tracked a predefined trajectory with an impedance controller [43]. The participants were instructed to intervene only once during the robot's task execution, in order to record a single physical correction. The resulting trajectories and physical interaction u_H were saved for offline analysis. This setup enabled us to easily analyze the situational confidence of the robot as we changed the robot's hypothesis space.

Next, we ran our approximate inference algorithm using the recorded human interaction torques and robot joint angle information. We measured what $\hat{\beta}$ would have been for each interaction if the robot knew about a given subset of the features. By changing the subset of features for the robot, we changed whether any given human interaction was well *explained* to the robot's hypothesis space.

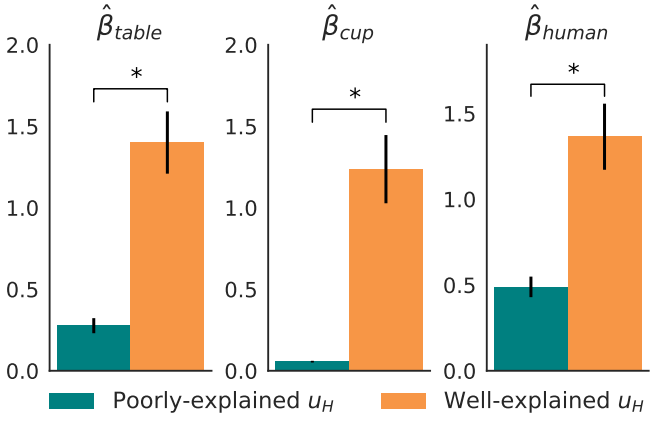
We tested two hypotheses:

H1. *Well-explained interactions result in high $\hat{\beta}$, whereas interactions that change a feature the robot **does not** know about result in low $\hat{\beta}$ for all features the robot **does** know about.*

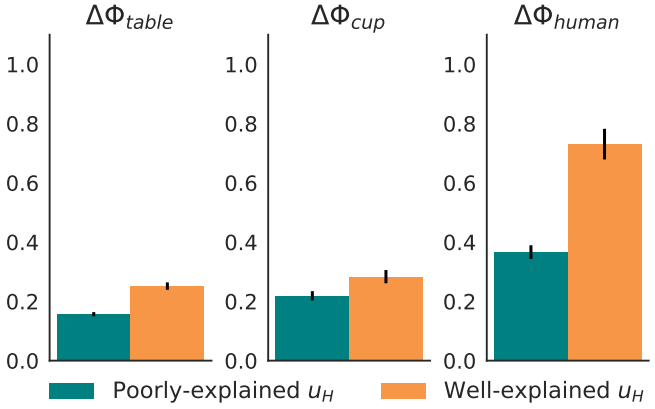
H2. *Not reasoning about well-explained interactions and, instead, indiscriminately learning from every update leads to significant unintended learning.*

We ran a repeated-measures ANOVA to test the effect of whether and input is well explained on our $\hat{\beta}$. We found a significant effect ($F(1, 521) = 9.9093$, $p = 0.0017$): when the person was providing a well-explained correction, $\hat{\beta}$ was significantly higher. This supports our hypothesis H1.

Fig. 11a plots $\hat{\beta}$ under the well-explained (orange) and poorly-explained (blue) conditions. Whereas the poorly-explained interactions end up with $\hat{\beta}$s close to 0, well-explained corrections have higher mean and take on a wider range of values, reflecting varying degrees of human performance in correcting something the robot knows about. We fit per-feature chi-squared distributions for $P(\hat{\beta} \mid E)$ for each value of E which we will use to infer E and, thus, θ online. In addition, Fig. 11b illustrates that even for poorly-explained human actions u_H, the resulting feature difference $\Delta\Phi = \Phi(\mathbf{x}_D) - \Phi(\mathbf{x})$ is non-negligible. This supports our



(a) Average β for well-explained and poorly-explained interactions.



(b) Average $\Delta\Phi$ for well-explained and poorly-explained interactions.

Fig. 11: β values are significantly larger for well-explained actions than for poorly-explained ones. Feature updates are non-negligible even during poorly-explained actions, which leads to significant unintended learning for fixed-β methods.

second hypothesis, H2, that not reasoning about how well-explained an action is is detrimental to learning performance when the robot receives misspecified updates.

VII. User Study on Learning from Corrections

Our case study on corrections suggested that $\hat{\beta}$ can be used as a measure of whether physical interactions are well explained and should be learned from. Next, we conducted an IRB-approved user study to investigate the implications of using these estimates during learning. During each experimental task, the robot began with a number of incorrect weights and participants were asked to physically correct the robot. Locations of the objects and human were kept consistent in our experiments across tasks and users to control for confounds[10]. The planning and inference were done for robot trajectories in 7-dimensional configuration space, accounting for all relevant constraints including joint limits and self-collisions, as well as

[10]We assume full observability of where the objects and the human are, as the focus of this paper is not sensing.

collisions between obstacles in the workspace and any part of the robots body.[11]

A. Experiment design

1) Independent variables: We used a 2 by 2 factoral design. We manipulated the corrections learning strategy with two levels (fixed-β and estimated-β learning), and also whether the human corrected for features inside (well explained) or outside (poorly explained) the robot's hypothesis space. In the fixed learning strategy, the robot updated its feature weights from a given interaction via (31) with a fixed β value. In the estimated-β learning strategy, the robot updates its feature weights via (30). The offline experiments above provided us an estimate for $P(E \mid \hat{\beta})$ that we used in the gradient update.

2) Dependent measures - objective: To analyze the objective performance of the two learning strategies, we focused on comparing two main measurements: the length of the $\hat{\theta}$ path through weight space as a measurement of the learning process, and the regret in feature space measured by $|\Phi(\mathbf{x}_{\theta^*}) - \Phi(\mathbf{x}_{actual})|$. Longer $\hat{\theta}$ paths should indicate a learning process that oscillates, whereas shorter paths suggest smoother learning curves. On the other hand, high regret implies that the learning method did not converge to a good objective θ, whereas low regret indicates better learning.

3) Dependent measures - subjective: For each condition, we administered a 7-point Likert scale survey about the participant's interaction experience (Table I). We separate the survey into 3 scales: task completion, task understanding, and unintended learning.

4) Hypotheses: We tested four hypotheses:

H1. *On tasks where humans try to correct **inside** the robot's hypothesis space (well-explained corrections), inferring situational confidence is not inferior to always assuming high situational confidence.*

H2. *On tasks where humans try to correct **outside** the robot's hypothesis space (poorly-explained corrections), inferring situational confidence reduces unintended learning.*

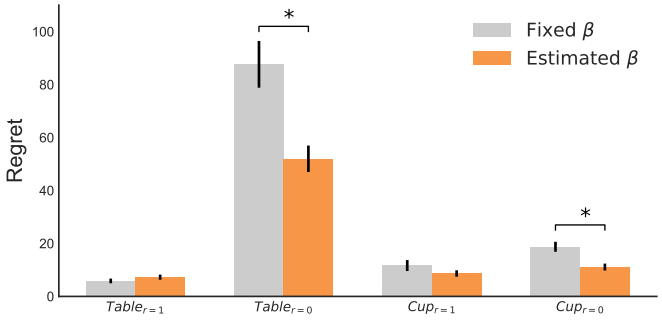
H3. *On tasks where they tried to correct **inside** the robot's hypothesis space, participants felt like the two methods performed the same.*

H4. *On tasks where they tried to correct **outside** the robot's hypothesis space, participants felt like our estimated-β method reduced unintended learning.*

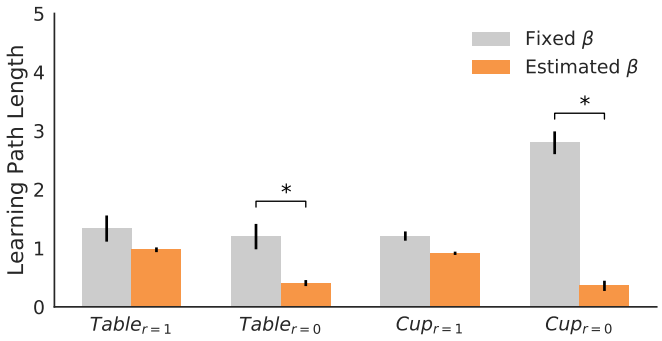
5) Tasks: We designed 4 experimental household motion planning tasks for the robot to perform in a shared workspace. Similarly to the case studies, for each experimental task, the robot carried a cup from a start to end pose with an initially incorrect objective. Participants were instructed to physically intervene to correct the robot's behavior during the task.

In Tasks 1 and 2, the robot's default trajectory took a cup from the participant and put it down on the table, but carried the cup too high above the table. In Tasks 3 and 4, the robot also took a cup from the human and placed it on the table, but this time it initially grasped the cup at the wrong angle, requiring human assistance to correct end-effector orientation to an upright position. For Tasks 1 and 3, the robot knew about the feature the human was asked to correct for ($E = 1$) and participants were told that the robot should be compliant. For Tasks 2 and 4, the correction was poorly explained ($E = 0$) and participants were instructed to correct any additional unwanted changes in the trajectory.

[11]For video footage of the experiment, see: https://youtu.be/stnFye8HdcU



(a) Regret averaged across subjects.



(b) $\hat{\theta}$ learning path length averaged across subjects.

Fig. 12: Comparison of regret and length of $\hat{\theta}$ learning path through weight space over time (lower is better).

6) Participants: We used a within-subjects design and randomized the order of the learning methods during experiments. We recruited 12 participants (6 females, 6 males, aged 18-30) from the campus community, 10 of which had technical background. None of the participants had experience interacting with the robot used in our experiments.

7) Procedure: Every participant was assigned a random ordering of the two methods, and performed each task without knowing how the underlying methods work. One challenge in performing and evaluating our experiment was that different participants may have different internal preferences for how a task should be performed. In order to have a consistent notion of ground-truth preferences, we fixed the true objective (e.g. how far the cup should be from the table) for each task. At the beginning of each task, the participant was first shown the incorrect default trajectory that they must correct, followed by the ground-truth desired trajectory they should teach the robot. This allows us to focus only on how well each algorithm infers objectives from human input, versus trying to additionally estimate the unique ground-truth human objective of each participant. Then the participant performed a familiarization round, followed by two recorded experimental rounds. After answering the survey, the participant repeated the procedure for the other method.

	Questions	Cronbach's α	F-Ratio	p-value
task	The robot accomplished the task in the way I wanted. The robot was NOT able to complete the task correctly.	0.94	0.88	0.348
understand	I felt the robot understood how I wanted the task done. I felt the robot did NOT know how I wanted the task done.	0.95	0.55	0.46
unintend	I had to undo corrections that I gave the robot. The robot wrongly updated its understanding about aspects of the task I did not want to change. After I adjusted the robot, it continued to do the other parts of the task correctly. After I adjusted the robot, it incorrectly updated parts of the task that were already correct.	0.91	9.15	**0.0046**

TABLE I: Results of ANOVA on subjective metrics collected from a 7-point Likert-scale survey.

B. Analysis

1) Objective: We ran a repeated-measures factorial ANOVA with learning strategy and input quality (well or poorly explained) as factors for the regret. We found a significant main effect for the method ($F(1, 187) = 7.8, p = 0.0058$), and a significant interaction effect ($F(1, 187) = 6.77, p = 0.0101$). We ran a post-hoc analysis with Tukey HSD corrections for multiple comparisons to analyze this effect, and found that it supported our hypotheses. On tasks where corrections were poorly explained, the estimated-β method had significantly lower regret ($p = 0.001$); on tasks where corrections were well explained, there was no significant difference ($p = 0.9991$). Fig. 12a plots the regret per task, and indeed the estimated-β method was not inferior on tasks 1 and 3, and significantly better on tasks 2 and 4.

For the length of the $\hat{\theta}$ path through weight space metric, the factorial ANOVA analysis found a significant main effect for the method ($F(1, 187) = 76.43, p < 0.0001$), and a significant interaction effect ($F(1, 187) = 33.3, p < 0.0001$). A similar post-hoc analysis with Tukey HSD correction for multiple comparisons also supports our hypotheses. On tasks where corrections were poorly explained, our method had significantly lower average weight paths over time ($p = 0.0025$); on tasks where correction were well explained, however, there was no significant difference ($p = 0.1584$). The same results are supported by Fig. 12b, which plots the average length of $\hat{\theta}$ through weight space per task, and indeed our method was not significantly inferior for tasks 1 and 3, and significantly better on tasks 2 and 4.

2) Subjective: We ran a repeated measures ANOVA on the results of our participant survey. We find that our method is not significantly different from the baseline in terms of task completion ($F(1, 7) = 0.88, p = 0.348$) and task understanding ($F(1, 7) = 0.55, p = 0.46$), which supports H3. Participants also significantly preferred the estimated-β method in terms of reducing unintended learning ($F(1, 7) = 9.15, p = 0.0046$), which supports H4.

VIII. Discussion

Human guidance is becoming increasingly important as autonomous systems enter the real world. One common way for robots to interpret human input is treating it as evidence about hypotheses in the robot's objective space. Since accounting for all possible hypotheses and situations ahead of time is challenging if not infeasible, in this paper we claim that robots should explicitly reason about how well their given hypothesis space can explain the human inputs.

We introduced the notion of *situational confidence* β as a natural way to measure how much the robot should trust its inputs and learn from them. We presented a general framework for estimating β in conjunction with any task objectives for scenarios where the human and the robot are operating the same dynamical system. We instantiated it for learning from human demonstrations, as well as for learning from corrections, by deriving a close to real-time approximate algorithm. In both settings, we exemplified – via human experiments with a 7-DoF robotic manipulator and a user study – that reasoning about situational confidence does, in fact, assist the robot in better understanding when it cannot explain human input.

There are several important limitations in our work. Perhaps the biggest limitation of all, which we alluded to in Section I, is that the hypothesis space can be misspecified but the robot can nonetheless explain the input relatively well, thus confusing misspecification for slight noise. This is especially true in more expressive hypothesis spaces, where there might always be some hypothesis that explains the input. This is unfortunately a fundamental problem with detecting misspecification in expressive hypothesis spaces: a single demonstration or a single data point will not be enough. Much like learning cost functions when using such spaces requires much more and diverse data than when using a less expressive space, with detecting misspecification too it will be the case that the robot will require a rich and diverse set of data points. The more data the robot has access to, and the more diversely it is distributed, the less of a chance there is that one wrong hypothesis can explain all the data.

Furthermore, our approach cannot disambiguate between misspecification of the hypothesis space and misspecification of the human observation model, i.e. the Boltzmann model.

Algorithmically, while for corrections we derived a way to handle continuous hypothesis spaces that scales linearly with the dimensionality of the space, for demonstrations we relied on simply discretizing the space. This was sufficient for showcasing the benefit of estimating situational confidence,

since for demonstrations this is done offline. However, to scale the method to complex spaces, we need to combine it with state-of-the-art (Bayesian) IRL approaches that rely on Metropolis Hastings sampling, or simply estimate the MLE.

Lastly, our experiments for both demonstrations and corrections are limited to a simple motion planning task with a cost function that depends on only a few features. We do not show how the method would degrade, both under ideal as well as under approximate inference.

In subsequent work, we hope to address some of these limitations. We are also interested in an extension to sequential time-dependent inputs, where the person could change their mind about what objective is important to them. Additionally, we want to explore ways of handling misspecification other than reducing learning, such as switching to a more expressive hypothesis space (but demanding more data and computation) whenever the situational confidence is very low for all θs. Finally, we are excited to showcase our work on other coupled dynamical systems, such as autonomous vehicles.

APPENDIX A
PRACTICAL CONSIDERATIONS

A. Demonstrations

1) Discretizing Θ and $\mathcal{B}$ in (13)*:* For the Θ discretization, we chose vectors in the unit sphere, as discussed in Section III-B. For practical purposes, we restricted the θ components to be positive due to our task features and the capabilities of our trajectory optimizer; in general, learning from demonstrations should be restricted to norm 1, not necessarily to the positive quadrant. In both our examples in Section IV and experiments in Sections VI, each θ_i component was allowed to take values 0, 0.5, or 1. Since we used 3 features, θ's dimensionality was 3, leading to a possible set Θ equivalent to the 3-fold Cartesian product of the values above. After normalizing to norm 1, we were left with 19 unique θ vectors in Θ, weighing the three features in different proportions, as shown in Figures 3, 7, 8, 9, and 10. Our discretization scheme ensured an approximately uniform sampling on the positive quadrant of the unit sphere.

To discretize situational confidence, we found it sufficient to cover $\beta \in \{0.01, 0.03, 0.1, 0.3, 1.0, 3.0, 10.0, 30.0, 100.0\}$, the log-scale space, similarly to [30], [31]. For different tasks, a similar discretization should suffice because what matters is β's relative magnitude for identifying misspecification, not its absolute one. We suggest calibrating the threshold ϵ in (6) using a few simulated trajectories like the ones in Fig. 3.

B. Corrections

1) Planning and Replanning: We use TrajOpt [41] to plan and replan robot trajectories. We set up the trajectory optimization problem to plan a path that minimizes a cost function of the form of (15). Given different features Φ and weights θ on these features, different optimal paths may be found. Additionally, we constrain the optimization to plan a path between a pre-specified start and goal locations, while avoiding collisions with the objects in the environment (table, laptop, or human). The total time of the trajectory is fixed, but the actual length can differ. That means that the robot moves faster for longer trajectories, and slower for shorter ones.

When the experiment starts, the robot plans an initial path from start to goal, using the initial weights θ. When a human push happens, the robot measures the instantaneous deviation, which deforms the trajectory via the impedance controller. Without learning, the robot would resume tracking its original trajectory. However, we use the human input to update θ according to (30), which the robot's planner uses to compute a new trajectory that the robot can follow instead. In a perfect world, this entire process would happen at 60Hz. In practice, however, the trajectory optimizer's computation lasts longer. As such, once a push is registered, the robot starts listening for following torque signals only after the update is complete.

Imagine this process in the context of a typical user experience. Once the person begins pushing, the robot instantly starts updating θ and optimizing the new induced path. While the person is applying their correction, the planner eventually finishes its computation and passes the updated trajectory to the robot controller. The user can immediately feel that the robot changed course and stops intervening.

2) Solving (21)*:* We used SLSQP, an off-the-shelf sequential quadratic programming package [44], to solve (21). In practice, the method can fail to return a good result if the initialization is bad. We found that if we initialize the minimization with a guess that does not satisfy the constraint (e.g. 0), it returns a reasonable estimate of the true u_H^*.

3) Sensitivity Analysis: Both (24) and (30) rely heavily on hyperparameters λ and ν. Here, we discuss how to set them.

Setting λ affects the magnitude of the resulting estimated situational confidence $\hat{\beta}$ in (24). This magnitude plays an important role when later estimating θ via (30) because it affects $P(E \mid \hat{\beta})$. However, note that to compute this probability we use $P(\hat{\beta} \mid E)$, which is an entirely data-driven empirical distribution, where the observed $\hat{\beta}$ is also computed via (24). As such, we are not relying on absolute magnitudes of the estimated situational confidence but on relative ones. Therefore, the choice of the hyperparameter λ does not affect our method's estimates as long as they are computed with the same hyperparameter that is used for learning $P(\hat{\beta} \mid E)$.

In the case of precision ν in (25), how spread out the Gaussian noise centered around $\Phi(\mathbf{x}_R)$ is affects the denominator in (30). When $\nu \to 0$, the $\Gamma(\Phi_D, E = 0)$ term in the denominator goes to 0, which means that (30) reduces to (31): our method always learns and never identifies misspecification. On the other hand, when $\nu \to \infty$, we can use the L'Hospital rule to see that $\Gamma(\Phi_D, E = 0) \to 0$ as well, as long as $||\Phi_D - \Phi(\mathbf{x}_R)||^2 \neq 0$, which is true unless there is no correction to deform $\mathbf{x}_R$, in which case we do not need to update θ at all. Therefore, it is important that ν is set not too high and not too low in order for our method to work properly.

The best practice for setting ν also involves using the offline data calibration from Section VI-B. To calibrate properly, after computing the empirical $P(\hat{\beta} \mid E)$ distribution, when $E = 0$ the updated θ should not change much, whereas when $E = 1$ the θ parameter should change appropriately.

Without the offline data calibration in Section VI-B, both λ and ν affect the θ and β estimation, and can have profound effects on the efficacy of our method. Unfortunately, we cannot do this calibration automatically yet, which is a limitation of our work, and we leave it for future research.

4) Trajectory Deformation Parameter Choice: When deforming the robot's trajectory given a human interaction, there are many choices of the deformation matrix A and the deformation magnitude parameter μ. A can be an explicit design choice (for example, constructing A from a finite differencing matrix [13]), can be solved for via an optimization problem which penalizes the undeformed trajectory's energy, the work done by the trajectory deformation to the human, and variations total jerk as in [45], or can even be learned from human data [46]. The magnitude of the deformation μ can also be tuned for best performance, for example to be robust to the rate at which deformations occur (see [27] for more details).

APPENDIX B
LAPLACE APPROXIMATION IN EQUATION (19)

Let the cost function in the model in (19) be denoted by:

$$C_{\Phi_D}(\bar{u}) = \lambda\|\bar{u}\|^2 + \kappa\|\Phi(\bar{\mathbf{x}}_D) - \Phi_D\|^2, \tag{32}$$

for an observed Φ_D.

First, our cost function can be approximated to quadratic order by computing a second order Taylor series approximation about the optimal human action u_H^* (obtained via the constrained optimization in 21):

$$\begin{aligned} C_{\Phi_D}(\bar{u}) \approx\ & C_{\Phi_D}(u_H^*) + \nabla C_{\Phi_D}(u_H^*)^\top(\bar{u} - u_H^*) \\ & + \frac{1}{2}(\bar{u} - u_H^*)^\top \nabla^2 C_{\Phi_D}(u_H^*)(\bar{u} - u_H^*) \ . \end{aligned} \tag{33}$$

Since $\nabla C_{\Phi_D}(\bar{u})$ has a global minimum at u_H^* then $\nabla C_{\Phi_D}(u_H^*) = 0$ and the denominator of Equation 19 can be rewritten as:

$$\begin{aligned} &\int_{\mathcal{U}} e^{-\beta C_{\Phi_D}(\bar{u})} d\bar{u} \approx \\ &\quad e^{-\beta C_{\Phi_D}(u_H^*)} \int_{\mathcal{U}} e^{-\frac{1}{2}(\bar{u}-u_H^*)\beta\nabla^2 C_{\Phi_D}(u_H^*)(\bar{u}-u_H^*)} d\bar{u} \ . \end{aligned} \tag{34}$$

Since $\beta\nabla^2 C_{\Phi_D}(u_H^*) > 0$ for $u_H^* \neq 0$, the integral is in Gaussian form, which admits a closed form solution:

$$\int_{\mathcal{U}} e^{-\beta C_{\Phi_D}(\bar{u}_H)} d\bar{u}_H \approx e^{-\beta C_{\Phi_D}(u_H^*)} \sqrt{\frac{2\pi^k}{\beta^k |H_{u_H^*}|}} \ ,$$

where $H_{u_H^*} = \nabla^2 C_{\Phi_D}(u_H^*)$ denotes the Hessian of C_{Φ_D} at u_H^*. Replacing $C_{\Phi_D}(\bar{u}_H)$ with the expanded cost function, we arrive at the final approximation of the observation model:

$$\begin{aligned} &P(u_H^t \mid x^0, \mathbf{u}_R; \Phi_D, \beta) \approx \\ &\qquad \frac{e^{-\beta\lambda(\|u_H^t\|^2)}}{e^{-\beta(\lambda\|u_H^*\|^2 + \kappa\|\Phi(\mathbf{x}_D^*) - \Phi_D\|^2)}} \sqrt{\frac{\beta^k |H_{u_H^*}|}{2\pi^k}} \ . \end{aligned} \tag{35}$$

ACKNOWLEDGMENT

This research is supported by the Air Force Office of Scientific Research (AFOSR) and the Open Philanthropy Project.

REFERENCES

[1] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.

[2] J. Z. Kolter, C. Plagemann, D. T. Jackson, A. Y. Ng, and S. Thrun, "A probabilistic approach to mixed open-loop and closed-loop control, with application to extreme autonomous driving," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 839–845.

[3] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation."

[4] F. Kpf, J. Inga, S. Rothfu, M. Flad, and S. Hohmann, "Inverse reinforcement learning for identification in linear-quadratic dynamic games," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 902 – 14 908, 2017, 20th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896317334596

[5] J. Fu and U. Topcu, "Pareto efficiency in synthesizing shared autonomy policies with temporal logic constraints," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 361–368.

[6] 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.

[7] A. Bobu, A. Bajcsy, J. F. Fisac, and A. D. Dragan, "Learning under misspecified objective spaces," in *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*, 2018, pp. 796–805. [Online]. Available: http://proceedings.mlr.press/v87/bobu18a.html

[8] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469 – 483, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889008001772

[9] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Machine Learning (ICML), International Conference on*. ACM, 2004.

[10] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters *et al.*, "An algorithmic perspective on imitation learning," *Foundations and Trends in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[11] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," *arXiv preprint arXiv:1503.07619*, 2015.

[12] A. Jain, S. Sharma, T. Joachims, and A. Saxena, "Learning preferences for manipulation tasks from online coactive feedback," *The International Journal of Robotics Research*, vol. 34, no. 10, pp. 1296–1313, 2015.

[13] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning robot objectives from physical human interaction," in *CoRL*, 2017.

[14] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," 06 2017.

[15] J. Fu, A. Singh, D. Ghosh, L. Yang, and S. Levine, "Variational inverse control with events: A general framework for data-driven reward definition," *arXiv preprint*, vol. arXiv:1805.11686.

[16] D. Hadfield-Menell, S. Milli, P. Abbeel, S. J. Russell, and A. D. Dragan, "Inverse reward design," in *NIPS*, 2017.

[17] R. E. Kalman, "When Is a Linear Control System Optimal?" *Journal of Basic Engineering*, vol. 86, no. 1, pp. 51–60, mar 1964. [Online]. Available: http://dx.doi.org/10.1115/1.3653115

[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] P. Shivaswamy and T. Joachims, "Coactive learning," *Journal of Artificial Intelligence Research*, vol. 53, pp. 1–40, 2015.

[20] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 729–736.

[21] M. Karlsson, A. Robertsson, and R. Johansson, "Autonomous interpretation of demonstrations for modification of dynamical movement primitives," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 316–321.

[22] R. Gutierrez, V. Chu, A. L. Thomaz, and S. Niekum, "Incremental task modification via corrective demonstrations," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1126–1133, 2018.

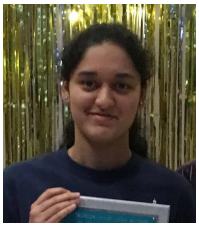
[23] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning from physical human corrections, one feature at a time," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '18. New York, NY, USA: ACM, 2018, pp. 141–149. [Online]. Available: http://doi.acm.org/10.1145/3171221.3171267

[24] C. Celemin, J. Ruiz-del Solar, and J. Kober, "A fast hybrid reinforcement learning framework with human corrective feedback," *Autonomous Robots*, vol. 43, no. 5, pp. 1173–1186, Jun 2019. [Online]. Available: https://doi.org/10.1007/s10514-018-9786-6
[25] B. D. Argall, E. L. Sauser, and A. G. Billard, "Tactile guidance for policy adaptation," *Found. Trends Robot*, vol. 1, no. 2, pp. 79–133, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1561/2300000012
[26] D. S. Brown, Y. Cui, and S. Niekum, "Risk-aware active inverse reinforcement learning," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87. PMLR, 29–31 Oct 2018, pp. 362–372. [Online]. Available: http://proceedings.mlr.press/v87/brown18a.html
[27] D. P. Losey and M. K. O'Malley, "Including uncertainty when learning from human corrections," *arXiv preprint arXiv:1806.02454*, 2018.
[28] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," *Urbana*, vol. 51, no. 61801, pp. 1–4.
[29] J. Zheng, S. Liu, and L. M. Ni, "Robust bayesian inverse reinforcement learning with sparse behavior noise," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI'14. AAAI Press, 2014, pp. 2198–2205. [Online]. Available: http://dl.acm.org/citation.cfm?id=2892753.2892857
[30] 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.
[31] 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.
[32] D. Hadfield-Menell, A. Dragan, P. Abbeel, and S. Russell, "Cooperative inverse reinforcement learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 3916–3924. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157382.3157535
[33] J. F. Fisac, M. A. Gates, J. B. Hamrick, C. Liu, D. Hadfield-Menell, M. Palaniappan, D. Malik, S. S. Sastry, T. L. Griffiths, and A. D. Dragan, "Pragmatic-pedagogic value alignment," *CoRR*, vol. abs/1707.06354, 2017.
[34] E. T. Jaynes, "Information theory and statistical mechanics," *Phys. Rev.*, vol. 106, pp. 620–630, May 1957. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.106.620
[35] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton University Press Princeton, NJ, 1945.
[36] C. L. Baker, J. B. Tenenbaum, and R. R. Saxe, "Goal inference as inverse planning," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 29, no. 29, 2007.
[37] 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
[38] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99 – 134, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000437029800023X
[39] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions," in *Robotics: Science and Systems*, 2017.
[40] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," *ArXiv*, vol. abs/1206.4617, 2012.
[41] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization."
[42] A. D. Dragan, K. Muelling, J. A. Bagnell, and S. S. Srinivasa, "Movement primitives via optimization," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2339–2346.
[43] N. Hogan, "Impedance control: An approach to manipulation: Part ii–implementation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16, Mar 1985. [Online]. Available: http://dx.doi.org/10.1115/1.3140713
[44] D. H. Kraft, "A software package for sequential quadratic programming," 1988.
[45] D. P. Losey and M. K. OMalley, "Trajectory deformations from physical human–robot interaction," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 126–138, 2017.
[46] H. J. Jeon and A. D. Dragan, "Configuration space metrics," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5101–5108.

Andreea Bobu received her B.S. degree in computer science and engineering from the Massachusetts Institute of Technology, United States in 2017. She is currently a Ph.D. student with the Interactive Autonomy and Collaborative Technologies Laboratory at University of California Berkeley in the United States. Her research interests include machine learning techniques for robot learning with uncertainty.

Andrea Bajcsy received her B.S. degree in computer science from the University of Maryland College Park, United States in 2016. She is currently a Ph.D. student in Electrical Engineering and Computer Sciences at the University of California Berkeley in the United States. She is the recipient of the National Science Foundation's Graduate Research Fellowship.

Jaime F. Fisac received a B.S./M.S. degree in Electrical Engineering from the Universidad Politcnica de Madrid, Spain, in 2012, the M.Sc. degree in Aeronautics from Cranfield University, U.K., in 2013, and the Ph.D. degree in Electrical Engineering and Computer Sciences from the University of California, Berkeley, U.S.A. in 2019. His research interests lie in control theory and artificial intelligence, with a focus on safety for autonomous systems. He is a recipient of the "la Caixa" Foundation Fellowship.



Sampada Deglurkar is a third year undergraduate student working towards her B.S. degree in Electrical Engineering and Computer Sciences at the University of California, Berkeley in the United States.

Anca Dragan received her Ph.D. degree in robotics from Carnegie Mellon University in 2015. She is currently an Assistant Professor in the Electrical Engineering and Computer Science department at the University of California Berkeley in Berkeley, United States.