Unified Learning from Demonstrations, Corrections, and Preferences during Physical Human–Robot Interaction

SHAUNAK A. MEHTA and DYLAN P. LOSEY, Virginia Tech, USA

Humans can leverage physical interaction to teach robot arms. This physical interaction takes multiple forms depending on the task, the user, and what the robot has learned so far. State-of-the-art approaches focus on learning from a single modality, or combine some interaction types. Some methods do so by assuming that the robot has prior information about the features of the task and the reward structure. By contrast, in this article, we introduce an algorithmic formalism that unites learning from demonstrations, corrections, and preferences. Our approach makes no assumptions about the tasks the human wants to teach the robot; instead, we learn a reward model from scratch by comparing the human's input to nearby alternatives, i.e., trajectories close to the human's feedback. We first derive a loss function that trains an ensemble of reward models to match the human's demonstrations, corrections, and preferences. The type and order of feedback is up to the human teacher: We enable the robot to collect this feedback passively or actively. We then apply constrained optimization to convert our learned reward into a desired robot trajectory. Through simulations and a user study, we demonstrate that our proposed approach more accurately learns manipulation tasks from physical human interaction than existing baselines, particularly when the robot is faced with new or unexpected objectives. Videos of our user study are available at https://youtu.be/FSUJsTYvEKU.

CCS Concepts: • **Computing methodologies** → **Online learning settings**; • **Human-centered computing** → **Collaborative interaction;**

Additional Key Words and Phrases: Physical human-robot interaction, reward learning, learning from multimodal feedback, imitation learning

ACM Reference format:
Shaunak A. Mehta and Dylan P. Losey. 2024. Unified Learning from Demonstrations, Corrections, and Preferences during Physical Human–Robot Interaction. *ACM Trans. Hum.-Robot Interact.* 13, 3, Article 39 (August 2024), 25 pages.
https://doi.org/10.1145/3623384

1 INTRODUCTION

Imagine teaching a robot arm on a factory floor (Figure 1). You know what task you want the robot to perform—attaching a chair leg—but you do not know how to program the robot to perform this task. Instead, you *physically* interact with the robot. Perhaps you start by kinesthetically guiding the robot across a full demonstration of the task. Next you deploy the robot, and notice it is attaching the chair leg at the wrong angle: Here you physically intervene and correct just that part of the arm's motion. Finally, as the robot gets closer to understanding your task, you might rank the

This work was funded in part by NSF Grant #2129201.
Authors' address: S. A. Mehta and D. P. Losey, Virginia Tech, Department of Mechanical Engineering, 635 Prices Fork Rd, Blacksburg, VA 24060; e-mails: {mehtashaunak, losey}@vt.edu.

2573-9522/2024/08-ART39
https://doi.org/10.1145/3623384

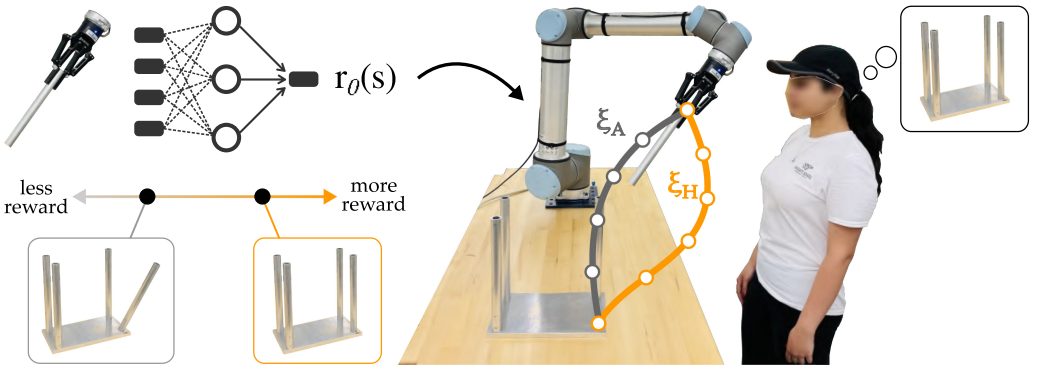

Fig. 1. Human teaching a robot arm to assemble a chair. The robot does not have any prior information about this task, and must learn from the human's physical interactions. We recognize that these interactions can take multiple different forms, including demonstrations, corrections, and preferences. To unify each type of input under a single framework, we train a reward model to assign higher scores to the human's behavior (ξ_R) than to nearby alternatives (ξ_A). The robot then optimizes this reward model to find its desired trajectory.

robot's behavior, indicating times where it gets it right and times where it messes up. Throughout this process, it does not make sense for you to be constrained to only a single type of interaction (e.g., only ever showing demonstrations of the task). Instead, you should be able to exploit *all* the avenues of physical human–robot interaction to convey your intent.

As robots increasingly share spaces with human partners, physical interaction between humans and robots becomes inevitable. Within this article, we focus on robot arms that perform manipulation tasks. Here, we can harness physical interaction as a channel for communication; when the human kinesthetically guides a robot arm throughout their desired task, the human is providing a *demonstration*; when the human pushes, pulls, or twists the robot to fix a part of its motion, the human is showing a *correction*; and when the human physically observes the robot and ranks its performance, the human is indicating their *preference*. Demonstrations, corrections, and preferences all communicate information about how the human wants the robot to behave. But we recognize that these modalities provide different—and often complementary—types of information: Demonstrations provide an outline of the high-level task, corrections hone-in on fine-grained aspects of the robot's motion, and preferences provide a ranked comparison between various repetitions of the same task. The right type of interaction (e.g., demonstration, correction, or preference) depends on the task, the user, and what the robot has learned so far.

Recent research on physical human–robot interaction develops separate approaches for each type of human feedback. Robot arms can learn manipulation tasks only from demonstrations [3], only from corrections [32], or only from preferences [23]. Moving beyond physical human–robot interaction, there is also work that unites combinations of these data sources: for instance, learning from demonstrations and preferences [4, 7, 9, 22, 49], or from demonstrations and corrections [20, 26, 42, 47]. But to learn from multiple sources of human feedback, the robot must either (a) have prior information about the tasks the human has in mind [4, 24] or (b) apply reinforcement learning to identify the optimal trajectory [7, 9, 10, 22, 49]. These constraints present practical challenges during physical human–robot interaction: Intuitively, we seek robots that learn new and unexpected tasks in real-time, without stopping to perform reinforcement learning through trial and error.

In this article, we propose a formalism for learning from physical interaction that unites demonstrations, corrections, and preferences. We recognize that these different types of interactions

provide different types of data about the human's desired task. To ground each feedback type within the same learning framework, our hypothesis is that:

We can unite physical demonstrations, corrections, and preferences under a single framework to learn a reward model that assigns higher scores to the human's input trajectories as compared to any modified alternatives.

Applying this insight we develop a two-step algorithm. First, we observe the human's physical interactions with the robot and learn a *reward model* from scratch. Second, we exploit the underlying structure of manipulation tasks by applying *constrained optimization* to map the learned reward into a robot trajectory. This process is iterative and free-form: At every iteration, the human can provide demonstrations, corrections, or preferences, and the robot updates its reward model and identifies an optimal trajectory in real-time. We emphasize that the reward model is a neural network that does not require any *a priori* knowledge about the human's potential rewards, and thus the current user is able to teach the robot arbitrary manipulation tasks. Returning to our running example, imagine that the robot has never assembled a chair before: But because the robot learns from the human's feedback to assign higher rewards to states where the chair legs are upright, the robot solves for a desired trajectory that carries the legs vertically.

Overall, we make the following contributions to physical human–robot interaction:

Uniting Physical Interactions. We present a learning approach to physical interaction that unifies demonstrations, corrections, and preferences. Our approach is based on the insight that the human's inputs are better than the alternatives: e.g., the human's demonstrated trajectory should receive higher reward than noisy perturbations of that same trajectory. Our learning approach automatically generates trajectory deformations of the human's physical interactions, and then learns to assign higher rewards to the human's actual behavior.

A Flexible Reward Learning Framework. We incorporate learning from both active and passive sources of feedback into a flexible reward learning framework. In a setting where the human may use multiple forms of feedback to teach the robot, we also enable the robot to actively prompt the human by eliciting their preferences. We identify a method for generating preference trajectories that—when ranked by the human—minimizes the robot's uncertainty over its learned reward. This approach results in an end-to-end model of the human's reward function. We then harness off-the-shelf optimization techniques and the robot's underlying kinematics to convert this learned reward function into a desired robot trajectory.

Comparing to Baselines. We compare our approach to state-of-the-art baselines across experiments with real robot arms and simulated and real human users. First, we consider approaches for physical human–robot interaction that learn from either demonstrations and corrections or demonstrations and preferences, and show that our method more accurately learns the human's task (particularly when this task is new or unexpected). Second, we perform a user study with imitation learning baselines that synthesize multiple forms of human feedback. Here, we show that participants prefer working with robots that learn using our approach, and that our approach learns trajectories that better align with the human's intended tasks. Readers can find videos of our experimental setup and user study at https://youtu.be/FSUJsTYvEKU.

2 RELATED WORK

This article introduces a learning formalism for physical human–robot interaction. During physical interaction, the human can kinesthetically guide the robot throughout examples of the task (demonstrations), apply forces and torques to adjust segments of the robot's motion (corrections), and rank the robot trajectories that they observe (preferences). Our approach seeks

to unite demonstrations, corrections, and preferences into a real-time learning algorithm. To enable safe and seamless human–robot interaction, we first take advantage of prior research on shared control. We then review two related topics from the existing literature: (a) learning methods designed specifically for physical human–robot interaction and (b) approaches outside of physical interaction that learn a reward function from multiple sources of human feedback.

Control Responses for Physical Interaction. Although we will focus on learning, we recognize that prior research has also explored control theoretic responses to physical interaction [12, 16, 17, 19, 31, 33, 38, 39]. Works on impedance control and shared control arbitrate leader-and-follower roles between the human and robot: When the human intervenes and applies large forces to the robot, the robot reduces its control feedback so that the human backdrives the arm. Intuitively, the robot can also pause, move away from the human, or follow the human's motion during physical interaction [16]. These control responses are an important step towards safety, and we will leverage shared control throughout this article to enable close physical interaction.

Learning Rewards during Physical Interaction. Recent work has enabled robots to learn from physical human–robot interaction [3]. Here, the human physically pushes, pulls, and twists the robot arm, and the robot attempts to infer why the human is applying these forces so that it can update its behavior accordingly. Some works directly map the human's forces and torques to changes in the robot's desired trajectory [2, 18, 27, 35]. But more relevant here is research that infers a reward function from physical interaction: These approaches assume that the human has in mind an objective, and the human's interactions are observations of this latent reward function [1, 40, 41]. The learned reward is then used to identify the robot's desired trajectory.

In practice, these approaches often take advantage of physical human corrections [6, 23, 25, 30, 32, 48]. Using the insight that the human's applied forces and torques are an intentional improvement—i.e., the human is going out of their way to show the robot a better way to perform the task—the robot learns to give the human's behavior higher rewards and propagate those changes the next time the robot repeats this task. In this article, we leverage a similar insight to derive our learning algorithm. However, while prior works for physical human–robot interaction focus on a single input modality (e.g., physical corrections), we will develop a framework that incorporates the different aspects of physical feedback.

Learning Rewards from Multiple Types of Feedback. Outside of physical interaction several methods have been proposed to learn from different types of human feedback. For example, interactive imitation learning approaches can synthesize demonstrations and corrections [20, 26, 42, 47]. Consider a human watching a mobile robot: First the human might teleoperate the robot throughout several laps of the building, and then the human may jump in and correct the robot only when it makes a specific mistake. Alternatively, methods that learn from suboptimal humans often combine demonstrations and preferences [7, 9, 10, 22, 29, 49]. Imagine that you are teaching a simulated robot to play an Atari game. After you do your best to provide a demonstration—and score as high as possible—you can rank the robot's autonomous performance to indicate when it is performing well and when it is making mistakes. Most relevant to our research are [24] and [4], where the authors unite different types of human feedback to learn a single reward model.

When applying these existing approaches to physical human–robot interaction, we are faced with two problems. On one hand, methods like [4, 24] require prior knowledge about the human's reward function—as we will show in our analysis and experiments, these methods fall short when the human wants to teach the robot a new or unexpected task. On the other hand, approaches like [7, 9, 10, 22, 29, 49] use reinforcement learning to convert the reward function into robot behavior. But reinforcement learning is time consuming and requires trial and error—which may not be possible (or safe) when humans are physically interacting with robot arms. Accordingly, in this

article, we introduce a formalism that unites demonstrations, corrections, and preferences without the need for pre-defined tasks or reinforcement learning.

3 PROBLEM STATEMENT

Going back to our motivating example from Figure 1, the human wants to teach their robot arm how to perform a manipulation task. To teach the robot the human exploits physical interaction: the human kinesthetically guides the robot through the process of attaching a chair leg, modifies specific sections of the robot's trajectory, and ranks the robot's autonomous behavior across repeated interactions. In this section, we formulate the problem of learning from each of these different types of physical interaction. We explain how existing approaches combine demonstrations, corrections, and preferences when the robot has prior knowledge about the tasks it will perform, and then highlight the shortcomings of these assumptions. Finally, we define our proposed reward model: We highlight that this approach can take advantage of the assumptions from previous works, but is also capable of learning new tasks that the robot does not know about beforehand.

Task. We formulate the task that the human wants the robot to perform as a Markov decision process: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, r, H \rangle$. Here, $s \in \mathcal{S}$ is the state of the robot and $a \in \mathcal{A}$ denotes the robot action. For instance, in our motivating example the robot's state s consists of its joint position and the position of objects in the scene, and the robot's action a is its joint velocity (e.g., a change in joint position). At timestep t , the robot transitions to a new state based on the dynamics $s_{t+1} = T(s_t, a_t)$. The task ends after a total of H timesteps.

Remember that—although the human knows what task the robot should perform—the robot may be uncertain about the correct task. We capture this objective using the reward function $r : \mathcal{S} \rightarrow \mathbb{R}$. The reward function maps robot states to scalar values (where higher scores indicate better states). To complete the task successfully, the robot must maximize the reward function; we will enable the robot to learn this reward function from physical interaction.

Trajectory. During each iteration of the task, the robot moves through a sequence of H states. We refer to this sequence as a trajectory $\xi \in \Xi$ such that $\xi = (s_0, s_1, \ldots s_H)$.

Demonstrations. The first way that the human can physically communicate with the robot arm is by providing demonstrations (Figure 2). During a demonstration, the human kinesthetically backdrives the robot throughout a trajectory ξ_d. We refer to the set of demonstrations as $\mathcal{D} = \{\xi_{d_1}, \xi_{d_2}, \ldots\}$, where each element of $\mathcal{D}$ is an H-length trajectory that shows the entire task.

Corrections. During demonstrations, the robot is passive throughout the entire interaction. But when the robot attempts to perform the task autonomously, the human may intervene to correct just a snippet of the robot's motion. Let $\xi_r \in \Xi$ be the trajectory that the robot is autonomously executing, and let $\xi_c \in \Xi_C \subset \Xi$ be the human's correction. We emphasize that ξ_c includes fewer than H timesteps, and the human only physically intervenes to push, pull, and guide the robot during ξ_c. Let the set of human corrections be written as $\mathcal{C} = \{(\xi_{r_1}, \xi_{c_1}), (\xi_{r_2}, \xi_{c_2}), \ldots\}$. Each correction consists of both the robot's initial trajectory and the human's modification.

Preferences. For demonstrations and corrections, the human is physically in contact with the robot arm. But the human and robot are also sharing the same space, and thus the human can observe the robot's trajectories and provide feedback about its behavior. Hence, the final way that the human conveys information to the robot is through preferences. We define a preference query as a set of k trajectories ranked in order by the human: $Q = \{\xi_1 \succ \xi_2 \succ, \ldots, \succ \xi_k\}$. Put another way, the human thinks ξ_1 better aligns with the task's reward than ξ_2. We group the human's preferences into a set $\mathcal{Q} = \{Q_1, Q_2, \ldots\}$, where each element of $\mathcal{Q}$ is a set of k ranked trajectories.

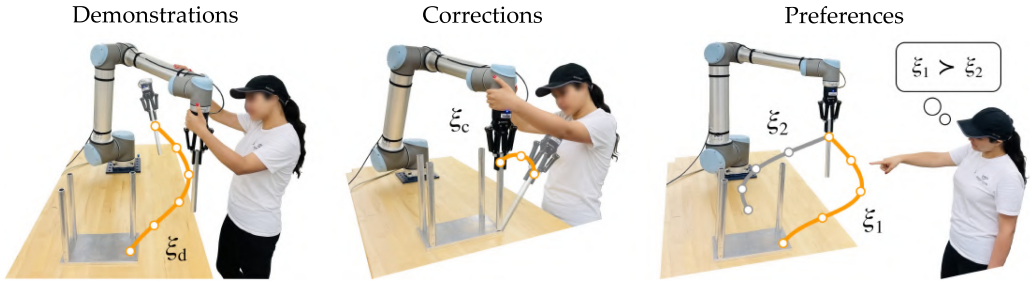

Fig. 2. Different types of physical feedback. (Left) Humans can convey information to robot arms by kinesthetically guiding the robot through a demonstration of the task. Demonstrations provide high-level information about the entire trajectory. (Middle) To refine a specific part of the robot's motion, humans may make physical corrections. These corrections fine-tune the robot's behavior. (Right) Over repeated interactions, the human will observe multiple robot trajectories. Humans can rank these trajectories (i.e., give their preferences) to indicate when the robot is making a mistake. We note that preferences are not physical—in the sense that the human does not apply forces or torques—but preference feedback naturally emerges when humans and robots occupy the same space and the human can physically observe the robot's behavior.

3.1 Preliminaries: Learning Rewards with Known Features

In order to formulate our problem, we first need to explain how other methods learn rewards from demonstrations, corrections, and preferences. Remember that we defined the reward function r as an arbitrary mapping from states to scores. Related works such as [4, 24, 32, 41, 50] introduce structural bias by assuming that the reward function is a linear combination of *features*:

$$r_\theta(s) = \theta \cdot \phi(s) \tag{1}$$

Here, $\phi : \mathcal{S} \rightarrow \mathbb{R}^n$ is the feature vector and $\theta \in \mathbb{R}^n$ is a weight vector. Features capture metrics that are potentially relevant to the current task. Within our running example from Figure 1, features could include the robot's distance from the chair, the angle of the chair leg, and the orientation of the chair. The robot is given these features *a priori* and must determine which features are important to the human; i.e., the robot is given ϕ and must learn θ. Sticking with our running example, the robot should learn to assign higher weight to the angle of the leg and lower weight to the orientation of the chair.

If we assume that the robot has access to the task-relevant features—and thus the reward is structured as in Equation (1)—we can use Bayesian inference to learn the correct weights θ. Let $P(\theta \mid \mathcal{D}, \mathcal{C}, \mathcal{Q})$ be the probability of weights θ given the human's previous demonstrations, corrections, and preferences. Applying Bayes' rule, and recognizing that each type of feedback is conditionally independent, we reach:

$$P(\theta \mid \mathcal{D}, \mathcal{C}, \mathcal{Q}) \propto P(\mathcal{D} \mid \theta) \cdot P(\mathcal{C} \mid \theta) \cdot P(\mathcal{Q} \mid \theta) \cdot P(\theta) \tag{2}$$

Here, $P(\theta)$ is the prior distribution over θ, and the $P(\cdot \mid \theta)$ terms capture the likelihood of the observed demonstrations, corrections, or preferences given that the human has reward weights θ. Previous works have found expressions for these likelihood functions [24]. For instance, we can model humans as approximately optimal, so that human inputs with higher rewards are increasingly likely [50]: $P(\xi_d \mid \theta) \propto \exp(\sum_{s \in \xi_d} \theta \cdot \phi(s))$. What is important here is that—if we assume access to the task-relevant features—inferring θ simplifies to Equation (2).

This preliminary approach makes sense if robots have access to all their reward features *a priori*. In practice, however, robots will inevitably face tasks they did not expect and features that were not pre-programmed [6]. Consider our motivating example: When we first bring this robot arm onto

the factory floor, will the robot understand the features of chair orientation or leg attachments? Human users should not be forced to hand-engineer features for each new task and environment; when the features are available, the robot arm should make use of their structure—but when the human's physical interactions are not aligned with any known feature, the robot should not be constrained to misspecified feature spaces [5]. Instead, we will develop robots that can learn reward functions *without* requiring predefined features.

3.2 Problem: Learning Arbitrary Rewards from Physical Interaction

Our goal is (a) to learn the task reward function from demonstrations, corrections, and preferences and then (b) to leverage this learned reward to identify an optimal robot trajectory that performs the task autonomously. To remove the reliance on features—and enable the robot to learn arbitrary task rewards—we will model the reward function as a neural network:

$$R_\theta(\xi) = \sum_{s \in \xi} r_\theta(s) \tag{3}$$

Here, θ is the weights of the neural network, $r_\theta(s)$ is the learned reward at state s, and $R_\theta(\xi)$ is the cumulative reward along trajectory ξ. If features are available we can incorporate them within this formulation. Define $s = (s, \phi(s))$ as an augmented state vector which now includes both the system state and the features $\phi(s)$. Returning to Equation (3), we learn a reward model r_θ that maps this augmented state to reward values: Cases where the task reward simplifies to $\theta \cdot \phi(s)$ are a special instance of our more general formulation.

We have chosen to learn a reward function because it provides an avenue to unify demonstrations, corrections, and preferences. In the next section, we will develop an algorithm to train this reward function from each different type of physical interaction.

4 UNIFYING DEMONSTRATIONS, CORRECTIONS, AND PREFERENCES

Our learning approach is based on comparisons. Recall that our underlying hypothesis is that the human's inputs—whether they are demonstrations, corrections, or preferences—are *intentional* improvements to the robot's behavior. Put another way, the reward model in Equation (3) should assign higher scores to human trajectories than to nearby alternatives. In this section, we apply our insight to develop a unified learning algorithm. First, we explain how to generate trajectory deformations that we can compare against the human's inputs. Next, we train the reward function to score the human's demonstrations, corrections, or preferences higher than these noisy alternatives. Finally, we leverage constrained optimization to convert our learned reward model into a robot trajectory. Throughout this section, we consider both passive communication (where the human chooses how to physically intervene) and active information gathering (where the robot prompts the human to uncover the correct reward function). We emphasize that our resulting approach is flexible, and humans can teach the robot using whichever physical feedback modalities they prefer.

4.1 Learning the Reward Model

Generating Trajectories for Comparison. Given an input trajectory ξ , we first seek to generate a nearby alternative $\hat{\xi}$. The intuition here is that the human chose to input ξ and not $\hat{\xi}$ — and thus the robot should assign higher rewards to ξ as compared to $\hat{\xi}$.

To create alternatives, we propagate noisy perturbations along the input trajectory following the approach outlined in [13, 34]:

$$\hat{\xi} = \xi + M\lambda \tag{4}$$

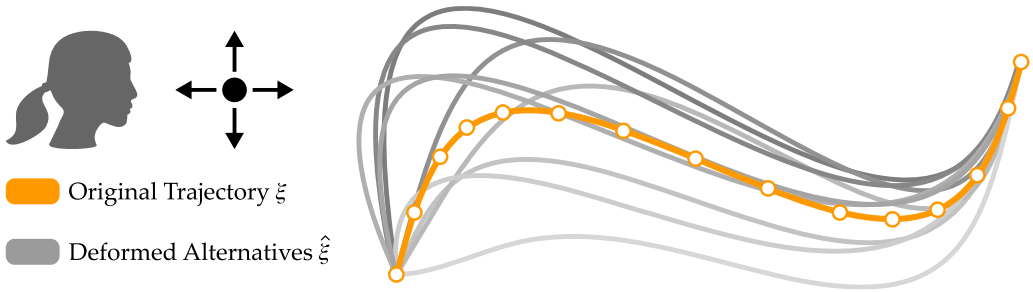

Fig. 3. Generating trajectories for comparison. In this example, the human moves a 2-DoF point mass robot along a sine wave. We record the initial trajectory ξ, and then apply Equation (4) to generate smooth perturbations $\hat{\xi}$. Our learned reward model should score ξ as a better trajectory than any of the alternatives $\hat{\xi}$.

Here, $\lambda \in \mathbb{R}^H$ is a noise vector that the designer selects and $M \in \mathbb{R}^{H \times H}$ is a symmetric positive definite matrix that defines a norm on the trajectory space. Our approach is not tied to any specific choice of M or λ; however, for our experiments, we selected the acceleration norm [13, 34] in order to get *smooth* trajectory deformations of ξ:

$$M = (A^T A)^{-1}, \quad A = \begin{bmatrix} 1 & 0 & 0 & & 0 \\ -2 & 1 & 0 & & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & & 0 \\ 0 & 0 & 1 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & & 1 \\ 0 & 0 & 0 & \cdots & -2 \\ 0 & 0 & 0 & & 1 \end{bmatrix} \tag{5}$$

Each deformation uses the same M and a new vector λ. In our experiments, we sampled λ from a zero-mean Gaussian distribution, and for every sampled λ we generated the corresponding $\hat{\xi}$ —to visualize this process, we show an example ξ and the generated alternatives $\hat{\xi}$ in Figure 3.

Learning from Demonstrations. Now that we have a way to generate nearby trajectories, we will learn the reward function by comparing these alternatives to the human's inputs. We start with demonstrations. Assuming that the human provides near-optimal demonstrations, the robot should learn to match each demonstration $\xi_d \in \mathcal{D}$. More formally, the robot should learn a reward model such that $R(\xi_d) > R(\hat{\xi}_d)$, where $\hat{\xi}_d$ is a deformation found using Equation (4).

Let $P(x \succ y \mid \theta)$ be the likelihood—from the robot's perspective—that trajectory x has higher total reward than trajectory y given that the robot has learned reward weights θ. Inspired by prior work on human decision making and Luce's choice axiom [36, 37, 46], we write this probability as a softmax-normalized distribution:

$$P(\xi_d \succ \hat{\xi}_d \mid \theta) = \frac{\exp R_\theta(\xi_d)}{\exp R_\theta(\xi_d) + \exp R_\theta(\hat{\xi}_d)} \tag{6}$$

Remember that we want the robot to assign higher rewards to ξ_d as compared to $\hat{\xi}_d$. When this happens we have that $P(\xi_d \succ \hat{\xi}_d \mid \theta) \to 1$ in Equation (6). Accordingly, to drive the probability $P(\xi_d \succ \hat{\xi}_d \mid \theta) \to 1$, we train the reward model to minimize the cross entropy loss [6, 7, 10, 22]:

$$\mathcal{L}_{\mathcal{D}}(\theta) = -\sum_{\xi_d \in \mathcal{D}} \mathbb{E}_{\hat{\xi}_d \sim \xi_d} \Big[\log P(\xi_d \succ \hat{\xi}_d \mid \theta) \Big] \tag{7}$$

Note that in Equation (7) we sum the cross entropy loss across every demonstration $\xi_d \in \mathcal{D}$, and for each demonstration we sample a set of nearby trajectories $\hat{\xi}_d$. Reward models that minimize Equation (7) will assign higher scores to trajectories that are like the human's demonstrations, and lower scores to trajectories that are different from these demonstrations.

Learning from Corrections. During demonstrations, the human backdrives the robot throughout the entire task; but during corrections, the human only fixes a snippet of the robot's trajectory. Given the robot's initial trajectory and the human's correction of this snippet—i.e., $(\xi_r, \xi_c) \in C$ —we recognize that $R(\xi_c) > R(\hat{\xi}_r)$, where $\hat{\xi}_r$ is the segment of the robot's trajectory that the human has intentionally modified. More generally, we assume that the human's correction shows the robot the right way to perform this part of the task. We therefore have $R(\xi_c) > R(\hat{\xi}_c)$, where $\hat{\xi}_c$ is a deformation of just the human's correction. Following the same derivation that we applied for demonstrations, we reach the following loss function for corrections:

$$\mathcal{L}_C(\theta) = \sum_{(\xi_r, \xi_c) \in C} -\log \frac{\exp R_\theta(\xi_c)}{\exp R_\theta(\xi_c) + \exp R_\theta(\hat{\xi}_r)} - \mathbb{E}_{\hat{\xi}_c \sim \xi_c}\left[\log \frac{\exp R_\theta(\xi_c)}{\exp R_\theta(\xi_c) + \exp R_\theta(\hat{\xi}_c)}\right] \tag{8}$$

Minimizing Equation (8) encourages the robot to learn a reward function that classifies ξ_c as a better trajectory than both the original segment $\hat{\xi}_r$ and perturbations $\hat{\xi}_c$ of the human's correction.

Learning from Preferences. As a final form of feedback the human operator can rank the robot's physical trajectories. Consider the robot arm in Figure 1: Each time the robot attempts to add a chair leg, the human onlooker might score the robot's performance, and mark whether trajectory ξ_i is better or worse than trajectory ξ_j. We emphasize that these preferences provide a *direct* comparison between pairs of trajectories. Here, we cannot assume that the human's preference is better than any other alternative; since the human's feedback only indicates that $\xi_i \succ \xi_j$, the reward model should learn $R(\xi_i) > R(\xi_j)$. Summing across the preferences $Q \in \mathcal{Q}$, we obtain the loss function:

$$\mathcal{L}_Q(\theta) = -\sum_{Q \in \mathcal{Q}} \sum_{(\xi_i \succ \xi_j) \in Q} \log \frac{\exp R_\theta(\xi_i)}{\exp R_\theta(\xi_i) + \exp R_\theta(\xi_j)} \tag{9}$$

where $(\xi_i \succ \xi_j)$ is any pair of human-ranked trajectories from preference Q. Intuitively, Equation (9) trains the reward model to rank the robot's trajectories in the same order as the human operator.

The human may passively provide these rankings over the course of repeated interactions. Alternatively, the robot can *actively* prompt the human by rolling out a set of trajectories and asking for the human's preference. The goal of these active prompts is to reduce the robot's uncertainty about the correct task reward. To formalize this uncertainty, we train an ensemble of m rewards with weights $\mathcal{E} = \{\theta_1, \theta_2, \ldots, \theta_m\}$. Each of these reward models is a separate instantiation of Equation (3) and is trained using the same demonstrations, corrections, and preferences. Let x and y be two robot trajectories. If all reward models agree on the relative scores of these trajectories—e.g., if $R_{\theta_i}(x) > R_{\theta_i}(y)$ for each $\theta_i \in \mathcal{E}$ —then the robot is confident that $x \succ y$. But in cases where the reward models disagree—for example, if $R_{\theta_1}(x) > R_{\theta_1}(y)$ while $R_{\theta_2}(x) < R_{\theta_2}(y)$ —then we do not know which trajectory is better at the task. At times when the reward models *disagree,* the robot needs additional human feedback to resolve its uncertainty.

To actively learn about θ , we will query the human by physically showing two robot trajectories, ξ_1 and ξ_2, and asking the human to choose which option they prefer. Humans may indicate that $\xi_1 \succ \xi_2$ or $\xi_2 \succ \xi_1$. Remember that we do not know beforehand how the human will respond to the

robot; accordingly, we select ξ_1 and ξ_2 such that—no matter which option the human chooses—the robot maximizes the information it gains about the unknown task reward θ:

$$(\xi_1^*, \xi_2^*) = \arg\max_{\xi_1, \xi_2 \in \Xi} \mathcal{I}(\theta; \xi_i \mid (\xi_1, \xi_2)) \tag{10}$$

Here, (ξ_1^*, ξ_2^*) is the greedily optimal query, ξ_i is the human's preferred trajectory, and $\mathcal{I}$ is the information gain [11]. Following the derivation in [4], we find that the expected information gain from query (ξ_1, ξ_2) across the ensemble $\mathcal{E}$ of reward models becomes:

$$\mathcal{I}(\theta; \xi_i \mid (\xi_1, \xi_2)) = \frac{1}{m} \sum_{\xi_i \in Q} \sum_{\theta \in \mathcal{E}} P(\xi_i \succ \xi_{-i} \mid \theta) \log_2 \left(\frac{m \cdot P(\xi_i \succ \xi_{-i} \mid \theta)}{\sum_{\theta' \in \mathcal{E}} P(\xi_i \succ \xi_{-i} \mid \theta')} \right) \tag{11}$$

where ξ_i is the trajectory the human prefers, ξ_{-i} is the other trajectory, and $P(\xi_i \succ \xi_{-i} \mid \theta)$ is the softmax-normalized distribution from Equation (6). In practice, Equation (11) looks for a query where (a) each reward model in the ensemble is confident about the relative scores of ξ_1 and ξ_2, but (b) some reward models think that $\xi_1 \succ \xi_2$, while other reward models think $\xi_2 \succ \xi_1$. We note that this active learning step is entirely optional. The robot still uses Equation (9) to learn from human preferences regardless of whether they are obtained passively or actively. However, as we will show in our experiments, actively selecting prompts using Equation (10) accelerates the robot's reward learning and resolves uncertainty across the ensemble of reward models.

Putting It All Together. We have identified loss functions that train the reward model to match the human's demonstrations, corrections, and preferences. For each of these interaction modalities, we have a common theme: The human's inputs should be scored higher than the alternatives. Our final step is to unite Equations (7)–(9) into a single loss function:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathcal{D}}(\theta) + \mathcal{L}_{C}(\theta) + \mathcal{L}_{Q}(\theta) \tag{12}$$

We note that Equation (12) is our parallel to the Bayesian inference from Equation (2).

We outline the implementation procedure for our approach in Algorithm 1. By controlling the number of comparisons for each feedback type (N_d, N_c , and N_p), we can adjust their relative weight. Within our experiments, we assign an equal importance to each of the feedback types. For our approach, the users can choose to provide any form of feedback to the robot by indicating their choice on a user interface. Previous work has proved that it is optimal to start with passive forms of feedback (e.g., demonstrations) before collecting active feedback (see Theorem 2 in [4]). Following this, the default order for our simulations and user study is demonstrations, then corrections, followed by preferences.

Given the human's inputs, we train an ensemble of reward models that minimize $\mathcal{L}(\theta)$. Each reward model is a fully connected network with two hidden layers and leaky rectified linear activation units. The output of the reward model is bounded between -1 and $+1$ using $\tanh(\cdot)$. Our ensemble includes three independently trained reward models: Each model optimizes its weights using the Adam learning rule with an initial learning rate of 0.001 [28]. To compute the reward of a state s , we take the average score from $r_{\theta_1}(s)$, $r_{\theta_2}(s)$, and $r_{\theta_3}(s)$. We retrain the reward models after each new demonstration, correction, or preference from the human.

4.2 Optimizing for Robot Trajectories

The first half of our formalism is learning a reward function (or ensemble of reward functions) from the human's physical interactions. In the second part of our approach, we convert this reward model r_θ into a robot trajectory ξ_r. Related approaches use reinforcement learning to identify the trajectory that maximizes r_θ [7, 9, 10, 22, 29, 49]; however, we recognize that reinforcement learning may not be appropriate for physical human–robot interaction. Here the human and robot

Algorithm 1: Learning from Multiple Forms of Feedback

Randomly initialize the ensemble of reward models with weights $\mathcal{E}^i = \{\theta_0^i, \theta_1^i, \cdots \theta_m^i\}$
Initialize the Demonstration, Correction and Preference buffers $\mathcal{D}, C, Q$
Initialize the number of noisy alternatives for $\mathcal{D}$, C and Q: N_d, N_c, N_q
for $i = 0, 1, 2, \cdots$ **do**
    Initialize the rankings buffer $\mathcal{B}$
    if $i = 0$ **and** Demonstration Provided **then**
        $\mathcal{D} \leftarrow \xi_d$
    else if Correction Provided **then**
        $C \leftarrow \xi_c$
    else if Preference Provided **then**
        $Q \leftarrow Q$
    end if
    for $j = 1, 2, \cdots N_d$ **do**
        Generate noisy alternative $\hat{\xi}_d^j$ for $\xi_d \in \mathcal{D}$ ▷ see Equation (4)
        $\mathcal{B} \leftarrow (\xi_d^j \succ \hat{\xi}_d)$
    end for
    for $j = 1, 2, \cdots N_c$ **do**
        Generate noisy alternatives $\hat{\xi}_c^j$ for $\xi_c \in C$ ▷ see Equation (4)
        $\mathcal{B} \leftarrow (\xi_c^j \succ \hat{\xi}_c)$
    end for
    for $j = 1, 2, \cdots N_q$ **do**
        Sample a preference $q^j \in Q$
        $\mathcal{B} \leftarrow q^j$
    end for
    Update the reward models $\mathcal{E}^i$
end for

are occupying the same space, and it becomes time consuming or unsafe for the robot to test multiple trajectories through the trial-and-error process of reinforcement learning.

Recall that our intended application is manipulation tasks for robot arms. Within this setting, we take advantage of the underlying robot kinematics to solve for the optimal robot trajectory. More formally, we leverage constrained optimization to convert the reward model into a robot trajectory:

$$\xi_r = \arg\max_{\xi \in \Xi} \sum_{\theta \in \mathcal{E}} \sum_{s \in \xi} r_\theta(s) \quad \text{s.t. } \xi(0) = s_0,\ \xi(H) = s_H \tag{13}$$

Here, s_0 is the start position of the robot arm (e.g., its current position) and s_H is a desired goal position. In practice, the goal position may not be known or there might not be a goal in the first place; in this case, we leverage Equation (13) without the constraint $\xi(H) = s_H$. Recent research on trajectory optimization has developed several approaches for Equation (13) [15, 21, 45]. Our formalism does not rely on any specific optimizer; in our experiments, we use sequential quadratic programming to solve Equation (13) and identify the optimal robot trajectory ξ_r.

Summarizing our Algorithm. At the start of the ith interaction, the robot has an ensemble of reward models with weights $\mathcal{E} = \{\theta_1, \theta_2, \ldots, \theta_m\}$. The robot applies Equation (13) to identify the optimal trajectory under the learned rewards, and then uses shared control to track this desired trajectory ξ_r. The human onlooker may intervene to kinesthetically guide the robot through the

task, physically correct the robot's motion, or rate the robot's overall behavior. We add this human feedback to the dataset of demonstrations $\mathcal{D}$ for the first interaction (i.e., if $i = 1$), and to the dataset corrections C, or preferences Q for all other interactions ($i > 1$). The robot then updates its reward models to minimize the unified loss function in Equation (12)—the robot leverages these updated rewards to the start of interaction $i + 1$.

5 SIMULATION 1: LEARNING FROM MULTIPLE FORMS OF INTERACTION

Now that we have developed a unified learning framework for physical human–robot interaction, we will compare different versions of our approach to the state-of-the-art baselines. As discussed in Section 2, several approaches learn from humans using physical interactions. Some of these approaches learn end-to-end models that capture the user's preferences for the task and learn a policy from the feedback provided, while others assume some knowledge over the features in the environment. In the latter, the features capture the task-specific concepts and are assumed to be prior knowledge of the tasks that the robot may need to perform. In this section, we perform a detailed comparison of various physical interaction approaches that learn from demonstrations, corrections and preferences, and their various combinations.

Independent Variables. We test 11 algorithms that learn from physical interactions. Among these 11 algorithms, 7 are different versions of Our approach, i.e., using only demonstrations (**Ours (D)**), only corrections (**Ours (C)**), only preferences (**Ours (P)**), demonstrations + corrections (**Ours (DC)**), demonstrations + preferences (**Ours (DP)**), corrections + preferences (**Ours (CP)**), and demonstrations + corrections + preferences (**Ours (DCP)**). We include three baselines that learn end-to-end networks without using any predefined features—human-gated **behavior cloning** (**BC**) [45], **adversarial inverse reinforcement learning** (**AIRL**), [14] and a method for learning from demonstrations and preferences developed for Atari games (**Atari**) [22]. We also use one baseline that assumes prior knowledge of the features in the environment and learns from a combination of demonstrations, corrections and preferences (**RRIC**) [24].

During **BC**, the robot learns a policy from the human's initial demonstrations. The robot then shows the trajectory to the human and the human can intervene to physically correct and improve the robot's behavior at any point in the trajectory. **AIRL** utilizes the demonstrations and corrections provided by the human to recover a reward function. The robot then optimizes that reward function to generate new behaviors, and compares them to the human's original inputs. We used the repository from [49] to implement **AIRL**. **Atari** uses a two-step approach. First, the robot leverages the human's demonstrations to learn a policy using imitation learning methods. The robot then shows the human sample trajectories generated using the learned policy, and the human indicates their preferences. These preferences are then used to learn a reward function that we optimize by applying the soft actor-critic algorithm and generating queries in **Atari**. Finally, in **RRIC,** the robot assumes full knowledge of the features in the environment and the reward function is modeled as a linear combination of these features. Based on the demonstrations, corrections and preferences provided by the human, feature weights are updated using Bayesian Inference. **Ours** directly learns a mapping from states to reward values and generates a trajectory to optimize that reward.

Procedure. The simulated human and a simulated robot performed two tasks (*Table* and *Laptop*) with each algorithm (Figure 4). For each of these tasks, the simulated human is teaching the robot to carry a cup of coffee to a goal position. In *Table*, the human wants the robot to carry the cup of coffee close to the table; in *Laptop*, the human wants the robot to avoid going over the laptop while moving to the goal location. For this experiment, we used a 7-DoF Franka-Emika Panda robot arm. The robot *did not have access* to the task reward function. The simulated human knew their reward

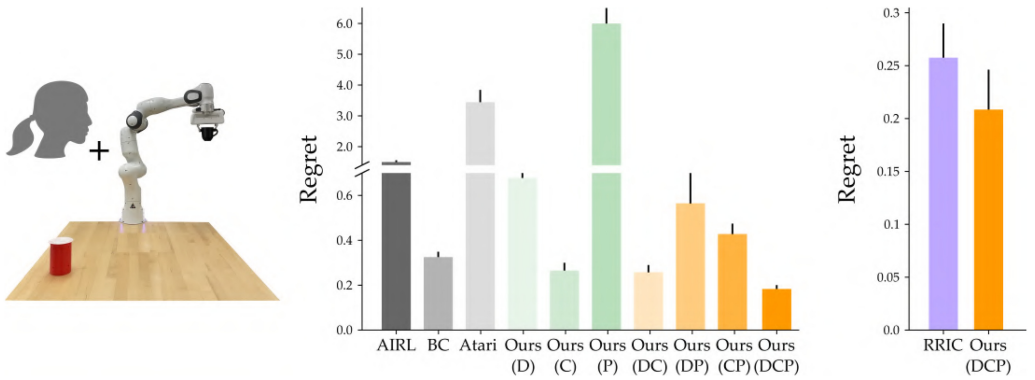

Fig. 4. Experimental results for simulated humans paired with a Franka Emika robot arm. (Left) We compare different versions of our approach to state-of-the-art end-to-end learning baselines as well as a feature-based approach that combines multiple forms of feedback. (Center) 15 simulated humans perform each task (*Laptop* and *Table*) using all the end-to-end learning algorithms. (Right) The simulated humans perform each task with a feature-based learning algorithm. We record the performance of the robot after learning from each approach in the form of regret and report the average regret and standard error. **Ours (DCP)** significantly outperforms all other versions of our approach ($p < .05$). **Ours (DCP)** has a significantly lower average regret as compared to the end-to-end learning methods ($p < .05$) and performs at par with **RRIC**. We emphasize that **RRIC** has access to all relevant features in the environment, while **Ours** learns the reward function from scratch.

function and provided demonstrations, corrections, and preferences to optimize that reward and teach the robot.

Within this experiment, we kept the number of interactions between the simulated human and the robot constant for each method. For **BC** and **AIRL**, the simulated human provided six demonstrations. For **Atari** , the human first provided two demonstrations and then asked for four preferences to the human. **RRIC** received an even split for each feedback type: two demonstrations, two corrections, and two preferences. Similarly, all different versions of **Ours** received an even split for each feedback type that version is meant to incorporate. For example, **Ours (D)** was given six demonstrations and **Ours (DC)** used three demonstrations and three corrections.

Dependent Variables. The simulated human worked with the robot to provide feedback across six interactions, and the robot optimized its learned reward to produce its final trajectory. We measured the performance of the robot by computing the *regret* of this learned trajectory:

$$Regret(\xi) = \sum_{s \in \xi^*} r_{\theta^*}(s) - \sum_{s \in \xi} r_{\theta^*}(s) \tag{14}$$

where r_{θ^*} is the true reward function for the task, ξ^* is the optimal robot trajectory for the task, and ξ is the robot's learned trajectory. Regret quantifies how much worse the robot's learned behavior is than the ideal behavior: Lower values of regret indicate better robot performance.

Hypothesis. For this simulation, we had the following hypotheses:

H1: *Our proposed approach with demonstrations, corrections, and preferences,* ***Ours (DCP)****, will outperform our approach with only one or two types of feedback.*
H2: ***Ours (DCP)*** *will outperform the end-to-end learning baselines.*
H3: *The performance of* ***Ours (DCP)*** *will match* ***RRIC*** *with known features.*

Results. Our results for this simulation are summarized in Figure 4. Performing a repeated measures ANOVA test (Normality of data verified using Q-Q plots) with a Greenhouse–Geisser correction, we found that the robot's learning algorithm had a significant effect on the regret ($F(1.818, 140) = 61.939, p < 0.05$). By contrast, we found that the task did not have a significant effect on the regret ($F(1, 14) = 3.073, p = 0.101$). Thus, we report the combined regret for both the tasks in our results.

From the regret plots in Figure 4 (Center), we observe that by combining all three forms of feedback our approach outperforms all other versions of **Ours** where we use only one or two forms of feedback ($p < 0.05$). This provides support for our hypothesis **H1**. We also observe that **Ours (C)** and **Ours (DC)** have a lower regret compared to **Ours (D)**, **Ours (P)**, **Ours (DP),** and **Ours(CP)**. This is a result of the iterative nature of the corrections as compared to the demonstrations, where the human provides all inputs at once before the robot updates its reward model.

We also notice that the end-to-end learning approaches perform at par with or better than some versions of **Ours** when only one or two types of feedback are available to our approach. While **Atari** and **Ours (DP)** receive the same forms of feedback, the regret for **Atari** is higher owing to the limited amount of data available for training. We observe that **BC** has a lower regret than **Ours (D)**, but performs at par with **Ours (C)**. This suggests that the performance of **Ours** improves when it receives incremental feedback from humans. However, when all three forms of feedback are made available to our approach, **Ours (DCP)** significantly outperforms all the end-to-end learning models ($p < 0.05$). This suggests that learning from multiple types of feedback is more effective than learning from just one or two types of feedback. Here, we find support for **H2**.

Finally, we compare **Ours (DCP)** to **RRIC**, which has access to the features in the environment and learns using all three forms of feedback within a Bayesian inference framework (Figure 4 (Right)). We observe that the performance of **Ours (DCP)** is comparable to **RRIC** ($p = 0.548$). This suggests that **Ours** —an approach that learns the reward end-to-end without any features— can perform as well as a Bayesian Inference approach that requires prior knowledge of all the features. Here, we find support for our hypothesis **H3**.

6 USER STUDY: MULTIPLE FORMS OF PHYSICAL INTERACTION

So far we have evaluated our method in controlled experiments with simulated human users. In this section, we will test our unified approach on *actual* participants. These participants physically interact with a 7-DoF robot arm by applying forces and torques, and communicate their intended task to the robot through demonstrations, corrections, and preferences. We compare our algorithm to interactive learning baselines that combine multiple types of human feedback. Here, we explore scenarios where the robot must learn the task from scratch: Our method and the baselines have no prior knowledge of the features or the tasks that the participants want to complete. Users must communicate their desired tasks through physical interaction. Videos of our user study are available at https://youtu.be/FSUJsTYvEKU.

Independent Variables. We tested four different algorithms that learn from physical interaction. Similar to Section 5, our baselines include human-gated **BC** [42], **AIRL** [14], and a method for learning from preferences and demonstrations developed for Atari games (**Atari**) [22]. **Ours** leverages the unified algorithm introduced in Section 4. We emphasize that each of these approaches learns without pre-defined features. The implementation of each of these approaches follows the procedure described in Section 5.

Experimental Setup. Participants physically interacted with a 7-DoF Franka Emika robot arm. During the user study, humans tried to teach this robot three tasks (Figure 5). In *Table,* the robot had to reach the goal while carrying a cup close to the table; in *Proximity,* the robot had to move to a goal region while staying away from the user. Note that the nature of these two tasks is similar

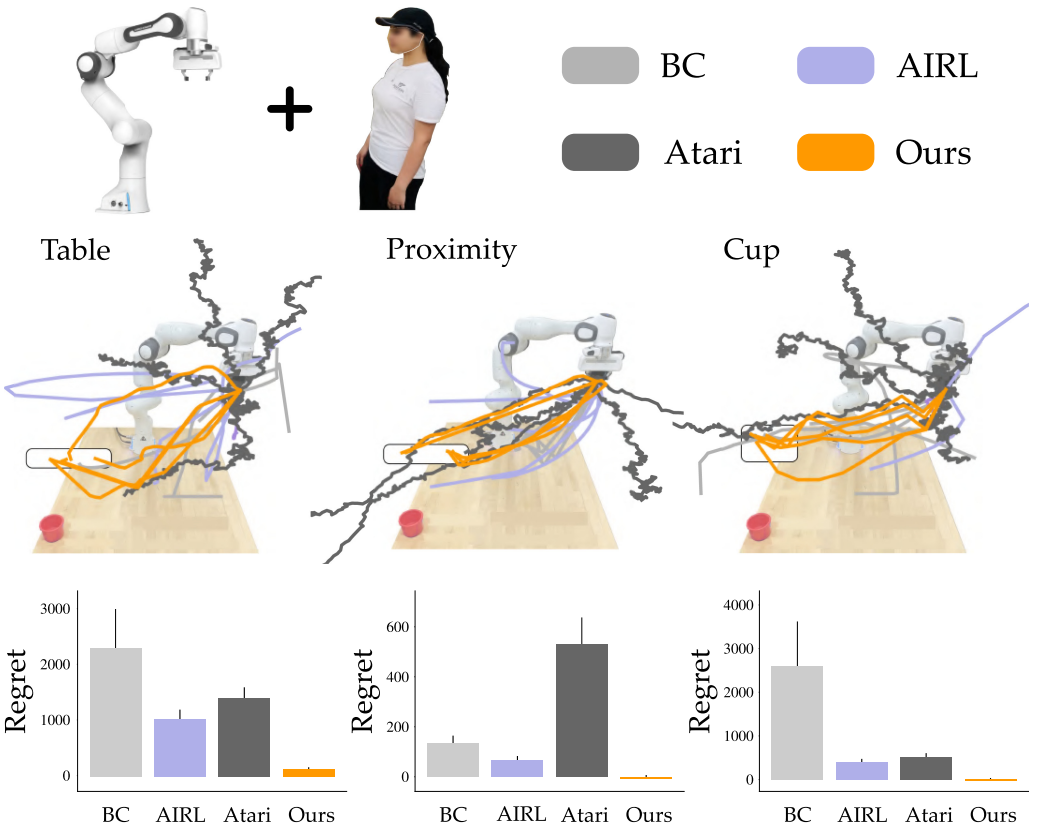

Fig. 5. Learned trajectories and objective results from our in-person user study. (Top) Participants physically interacted with a 7-DoF robot arm that had no prior knowledge about the tasks. The robot learned from physical interactions using our approach and imitation learning baselines that combine multiple feedback modalities. (Middle) The final trajectories the robot learned with each method. Five users taught the robot the *Table* task, five users taught the *Proximity* task, and five users taught the *Cup* task. During each task, the robot needed to reach a goal position within the white rectangle. We trace the xyz position of the robot's end-effector; within the *Cup* task, the robot also needed to maintain specific orientations. (Bottom) The regret between the robot's learned trajectory and ideal trajectory. Lower values of regret indicate that the robot completed the task correctly, and the error bars plot standard error of the mean. textbfOurs outperforms **AIRL** and **Atari** on the *Table* and *Cup* tasks ($p < .05$), and **Ours** has a lower regret than all the baselines for the *Proximity* task ($p < .05$).

to that of the experiments performed in Section 5. In this user study, we introduce a new task, *Cup*, where the participants teach the robot to complete a scooping action and then pour the cup at the goal position. We asked participants to mark their goal at the start of each interaction. To encourage more diverse human inputs, participants were instructed to change their goal position within a marked region between interactions.

Participants and Procedure. For our user study, we recruited 15 participants from the campus community (5 female, average age 25 ± 4 years). Participants gave informed written consent prior to the start of the experiment under Virginia Tech IRB #22-308.

The participants were divided into three groups of five people. Each group of participants performed a single task (i.e., participants only taught the table task, the proximity task, or the cup task).

Importantly, users taught this task with *all four* of the robot learning algorithms. The order of the algorithms was counterbalanced using a Latin square design: e.g., some participants began with **Ours**, and others began with **BC**. For each learning algorithm, the human and robot started over from scratch: The robot had no prior information, and the human provided new demonstrations, corrections, or preferences to convey their task.

For **BC** and **AIRL**, the human provided six demonstrations.[1] With **Atari** the participant first provided two demonstrations and then the robot asked for four preferences (to reach a total six interactions). Finally, observing the result from Section 5, that multiple forms of feedback enable a better learning, we provide our approach with all three forms of feedback. We divide **Ours** evenly between each type of physical feedback: Users gave two demonstrations, corrections, and preferences (to maintain a total of $2 + 2 + 2 = 6$ interactions).

Dependent Variables. After participants finished providing their inputs, the robot leveraged its learning algorithm to identify a final trajectory. We measured how effectively this final trajectory completed the intended task. More specifically, we applied 14 to quantify the regret between the robot's actual behavior and the ideal task behavior.

We also administered a 7-point Likert scale survey [44] to assess the participants' subjective responses to each learning condition. Our survey questions were organized into four multi-item scales: how *easy* it was to physically provide feedback to the robot, whether the robot *learned* to perform the task correctly, how *flexible* the robot was to different types of physical interaction, and if they would *prefer* using this method in the future. Every participant completed this survey four times: once after they finished working with each robot learning algorithm.

Hypothesis. For our user study, we had the following hypotheses:

> **H4:** *Robots using our unified learning approach will perform the task better after the same number of physical human interactions.*
> **H5:** *Participants will subjectively prefer our learning algorithm as compared to the baselines.*

Results. The objective results from our user study are presented in Figure 5, and the subjective responses are summarized in Figure 6. Let us start with the objective results: After verifying the normality of the data using Q-Q plots and performing a repeated measures ANOVA test on our results, we found that the robot's learning algorithm had a significant main effect on regret ($F(3, 12) = 6.942, p < .05$). Looking at Figure 5 we notice that the regret for **Ours** is consistently lower than the baselines. Here, lower regret is better—this indicates that the robot learned trajectories better matched the human's intended task. We can directly observe this trend from the final trajectories shown above: Notice that **Ours** consistently moves to the goal region and completes the task, while the alternatives produce noisy, inconsistent motions. Post hoc comparisons confirm that **Ours** outperformed **AIRL** and **Atari** on the *Table* and *Cup* tasks ($p < .05$), and that **Ours** had an overall lower regret than all baselines for *Proximity* task ($p < .05$). We observe that given the limited amount of data (only six interactions), **Ours** has lower average regret with a low variance across all three tasks. On the other hand, **Atari** and **BC** fail to learn the task representations accurately from the same limited amount of data, and thus have a higher variance in their performance. Overall, the results shown here and in the video submission indicate that our unified learning approach was best able to synthesize the human's physical inputs and learn the correct task from scratch.

[1]For **BC** and **AIRL** , we gave users the option of providing corrections that only modify a segment of the robot's behavior. However, since the robot's learned behavior after two demonstrations was far from the user's intended task, participants chose to keep demonstrating the entire trajectory. Note that neither **BC** or **AIRL** can learn from preferences.

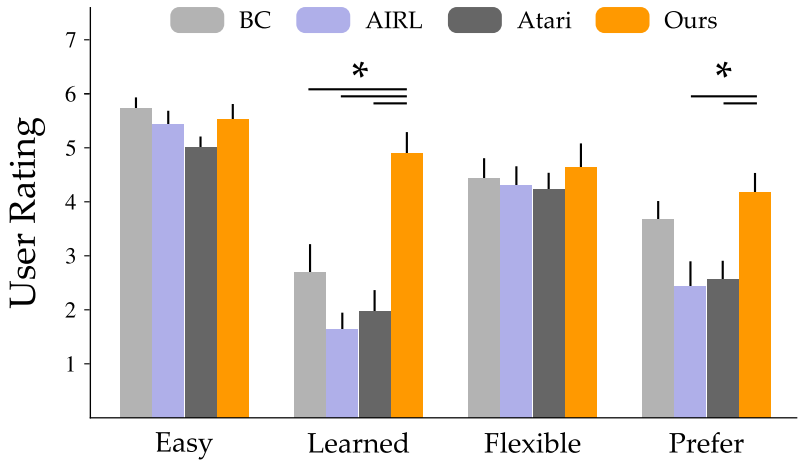

Fig. 6. Subjective results from our in-person user study. Higher ratings indicate user agreement (e.g., a score of 7 indicates that it was *easier* to provide physical feedback). Error bars show standard error of the mean, and an $*$ denotes statistical significance ($p < .05$). After watching the final trajectory learned by each approach, participants rated **Ours** as a better *learner* than the baselines. Users also *preferred* **Ours** to **AIRL** and **Atari**.

Next we consider the results from our Likert scale survey in Figure 6. After confirming that the scales were reliable (Cronbach's $\alpha > 0.7$), we grouped each scale into a single combined score and performed a one-way repeated measures ANOVA on the results Figure 7 and Table 1. When users watched the robot's final learned behavior they perceived **Our** approach as a better learner than the baselines ($F(3, 42) = 11.982, p < .05$), and when they considered their experience teaching the robot they preferred to use **Ours** over the **AIRL** and **Atari** approaches ($F(3, 42) = 4.263, p < .05$). One confounding factor here is the about of time it took for the robot to learn from human's demonstrations. With **Ours** and **BC** the entire process from teaching the robot to autonomously completing the task took roughly 10 minutes, while with **AIRL** and **Atari** it took more than 15 minutes on average (this additional time was needed to train the robot's policy). Participants may have preferred **Ours** and **BC** in part because they completed the training process more quickly. However, our overall results support hypothesis **H5** and suggest that participants subjectively perceived our unified approach as a better learner from demonstrations, corrections, and preferences.

7 SIMULATION 2: LEARNING WITH KNOWN AND UNKNOWN FEATURES

Now that we have tested our approach against baselines that learn end-to-end models, we will compare our approach to state-of-the-art baselines that use the knowledge about the features in the environment to learn the reward weights. As we discussed in Section 3.1, several related works learn from combinations of demonstrations, corrections, and preferences by assuming that the reward function is based on features. These features capture task-relevant concepts (e.g., the orientation of the chair leg) and are programmed using prior knowledge of the tasks the robot will need to perform. Here, we consider situations in which the robot must learn *expected* tasks—i.e., cases where the features apply—as well as *unexpected* tasks where the pre-programmed features are insufficient. We conduct these experiments with simulated humans that provide inputs to real robot arms. Overall, we break this section down into two parts: (a) a comparison to physical interaction approaches that learn from demonstrations and corrections, and (b) a comparison to learning approaches that combine demonstrations and preferences.

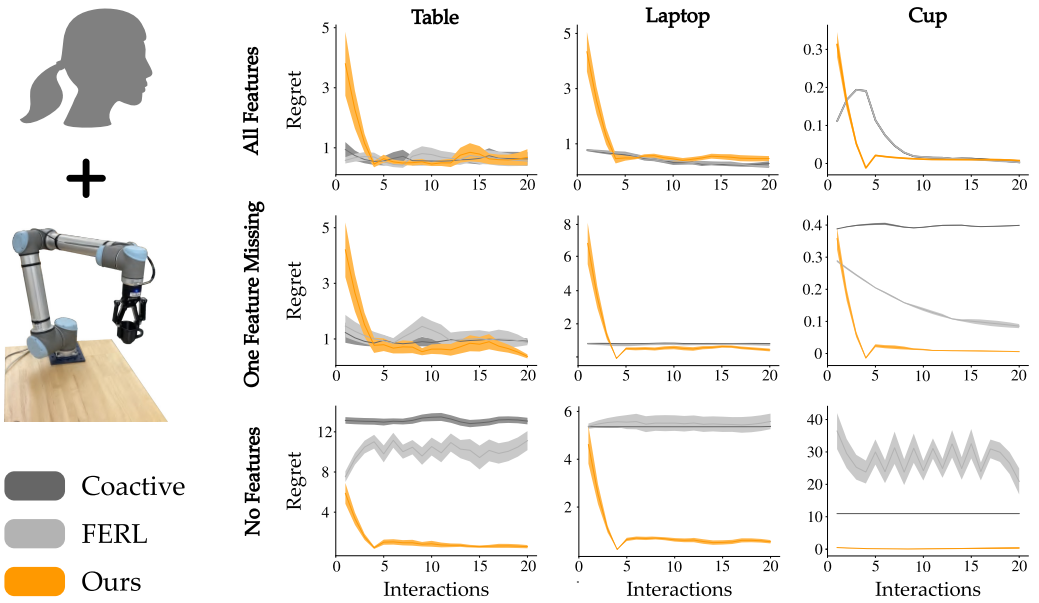

Fig. 7. Experimental results for simulated humans paired with a UR10 robot arm. (Left) We compare our approach to two existing algorithms that learn from physical interaction. **Coactive** [23, 32] assumes that the reward is composed of pre-programmed features, and **FERL** [6] learns features from human demonstrations before constructing a reward function from those features. (Right) Over repeated interactions, 10 simulated humans input demonstrations and corrections to teach the *Table*, *Laptop*, and *Cup* tasks. The columns correspond to the tasks, and the rows capture the prior information the robot has about each task. In the first row the robot is given all task-related features, in the middle row the robot is missing one feature, and in the bottom row the robot has no prior information about the task. The plots show regret (the difference in reward between the ideal trajectory and the learned trajectory), and the shaded regions show the standard error. At the end of all 20 interactions, **Ours** performs similar to or worse than **Coactive** and **FERL** when all features are known. If one or more feature is missing, however, **Ours** outperforms both baselines.

Table 1. Questions on Our Likert Scale Survey

Questionnaire Item	Reliability	F(3,42)	p-value		
			BC	AIRL	Atari
–It was easy to provide feedback to the robot. –Providing feedback to the robot was challenging.	0.863	1.707	0.486	0.7	0.185
– The robot learned to perform the task correctly. – The robot's motion did not align with what I was trying to teach the robot.	0.911	11.982	$p < 0.05^*$	$p < 0.05^*$	$p < 0.05^*$
–I liked showing different types of feedback. –I preferred just showing one type of feedback repeatedly.	0.789	0.225	0.689	0.571	0.427
– I would use this method in the future. – I would prefer another approach that I tried if I was to do this again.	0.806	4.263	0.353	$p < 0.05^*$	$p < 0.05^*$

We grouped questions into four scales and tested their reliability using Cronbach's α. We explored whether providing feedback to the robot was easy, the robot learned the task, if the users liked the flexibility of using different feedback forms, and if they preferred to use the method in future. Computed p-values indicate if users preferred our approach to the baselines, where * denotes statistical significance.

7.1 Learning from Demonstrations and Corrections

Independent Variables. We consider two baselines for learning from physical demonstrations and corrections: **Coactive** [23, 32] and **FERL** [6]. In **Coactive** , the robot assumes that it knows *all* the features for the current task; based on the human's inputs, the robot builds a reward function from these features and then selects the optimal trajectory. By contrast, in **FERL** , the robot recognizes that it may be missing some task-related features. Here, the robot leverages the feature demonstrations (feature traces) provided by the human to first learn the unknown features—once it has a model of these features, it then applies the same method as **Coactive** to build the reward function. Remember that our proposed approach (**Ours**) does not rely on features. Instead, we learn a mapping directly from states to rewards; for cases where the features are given, **Ours** can incorporate those features in the augmented state to be used as an input to our reward model (for learning and execution). Note that here, our reward model has the same information about the features as the other approaches (i.e., if **Coactive** and **FERL** have information about only *1 feature*, **Ours** also has information about only one feature).

Procedure. The simulated human and real robot performed three tasks with each learning algorithm (Figure 7). For all three tasks, the human is teaching the robot to reach a goal position. In *Table,* the human wants the robot arm to move close to the table; in *Laptop,* the human wants the robot arm to avoid passing above a laptop; and in *Cup,* the human wants the robot arm to carry a cup upright so that it does not spill. Each task has two potential features: the goal that the robot should reach (e.g., distance to the goal) and the way the robot arm should move towards that goal (e.g., height from the table, distance from the laptop, or orientation of the cup). For these experiments, we used a 6-DoF UR10 robot arm. This robot *did not know* the task reward function. To teach the real robot in a controlled setting, we used a simulated human: The simulated human knew the correct reward, and provided demonstrations or corrections that optimized this reward.

We seek to understand how our approach compares to baselines both when the robot has prior knowledge about the task and when the task is new or unexpected. Accordingly, we tested three different conditions: (a) when the robot knows all task-related features, (b) when one task-related feature— *Laptop*, *Table,* or *Cup* —is missing, and (c) when the robot does not know any features of the task.

Dependent Variables. The simulated human worked with the real robot to provide 20 inputs (i.e., demonstrations and corrections) over repeated interactions. For **Coactive** and **Ours,** the first input is a task demonstration, and the remaining interactions are corrections. For **FERL** , the type of input depends on the number of unknown features. When all the features are given, **FERL** also starts with a task demonstration followed by corrections. But when any feature is missing, the human first provides 10 feature demonstrations per missing feature. After these offline demonstrations (which are meant to teach features to the robot), the human provides one task demonstration and corrections similar to **Coactive** and **Ours**. Thus, each method receives one task demonstration and 19 corrections to learn the task (the feature demonstrations are not included as part of the 20 interactions). After each interaction, we measured the performance of the learning robot using Equation (14).

Hypothesis. For this experiment, we had the following hypotheses:

> **H6:** *Our method will match the baselines when all the features are known.*
> **H7:** *Our method will outperform both **Coactive** and **FERL** when one or more features are missing.*

Results. Our results are visualized in Figure 7. Note that this figure is a grid: The columns are the tasks, and the rows are the amount of prior knowledge that the robot has about the tasks. To

analyze the results, we first verified the normality of our data using Q-Q plots and then performed a repeated measures ANOVA with Greenhouse–Geisser correction. We found that the robot's learning algorithm had an effect on regret across all tasks and conditions: $F(1.040, 18) = 29.063$, $p < 0.05$.

To understand this result, we next explored how the robot's performance changed based on the amount of prior information available to the learning algorithm. In the first row of Figure 7, we consider scenarios where all the task-related features are known. For example, during the *Table* task, the robot knows that the reward is a function of the distance from the goal and the height of the end-effector. Overall, we found that—when all features are given— **Ours** performs on par with or worse than the baselines by the end of all 20 interactions. Post hoc tests revealed that for the *Table* task, **Ours** matched the alternatives (**Coactive**: $p = 0.788$, **FERL**: $p = 0.790$). During *Laptop,* our method performed similarly to **Coactive** ($p = 0.117$) but had higher regret than **FERL** ($p < 0.05$). Finally, within *Cup* both **FERL** and **Coactive** outperformed our approach at the last interaction ($p < 0.05$) These results partially support hypothesis **H6**: When the robot is given prior information about all the features, existing methods leverage this structure to accurately learn the human's reward. Our approach starts with worse performance because it does not make assumptions about the reward function and must learn to focus on the given features.

However, the relative performance changes once the robot encounters new or unexpected tasks. In the second row of Figure 7, we test settings where the robot knows one feature (distance to the goal) but does not know the other task-related feature. Because **Coactive** assumes that all the features are given, it treats each human input as an observation about the correct distance to the goal (and never realizes that the human's inputs may be communicating something else). **FERL** takes a step towards resolving this problem by trying to learn the missing feature from the first four interactions. But post hoc tests reveal that our proposed learning approach matches or outperforms the feature-based alternatives. For the *Table* task, there are no statistically significant differences between **Ours** and **Coactive** ($p = 0.074$), but **Ours** has a lower regret than **FERL** ($p < 0.05$). On the other hand, during *Laptop* and *Cup,* our method leads to less regret than both the baselines by the end of the physical interactions ($p < 0.05$). This trend continues in the final row of Figure 7 where the robot has no prior information about the task reward. Here, **Ours** outperforms both baselines across all three tasks ($p < 0.05$). Overall, these results suggest that when the robot encounters situations where it has incomplete information, our unstructured reward learning approach is better able to capture the correct task than baselines that depend on task-related features. We therefore find support for **H7** when the robot is learning from physical demonstrations and corrections.

7.2 Learning from Demonstrations and Preferences

Independent Variables. So far we have compared our approach to baselines developed specifically for physical interaction when learning from demonstrations and corrections. Next, we turn our attention to alternate approaches that learn from demonstrations and preferences [7, 9, 22, 24, 49]: Although these methods are not designed only for physical interaction, they can be applied to our setting. Here, we compare **Ours** to **DemPref** [4], a recent approach that combines both demonstrations and preferences to build a model of the human's reward function. Like **Coactive** in the previous experiment, **DemPref** assumes that the reward function is composed of features, and the robot knows all the relevant features for the current task. In this experiment, we aim to study the tradeoff between our proposed approach and approaches that utilize Bayesian inference to learn the task representation from human feedback.

The **DemPref** algorithm has two parts. First, the robot gets demonstrations from the human to learn a rough estimate of the reward function; next, the robot actively queries the human to elicit their preferences and fine-tune the learned reward. Recall from Section 4 that under our

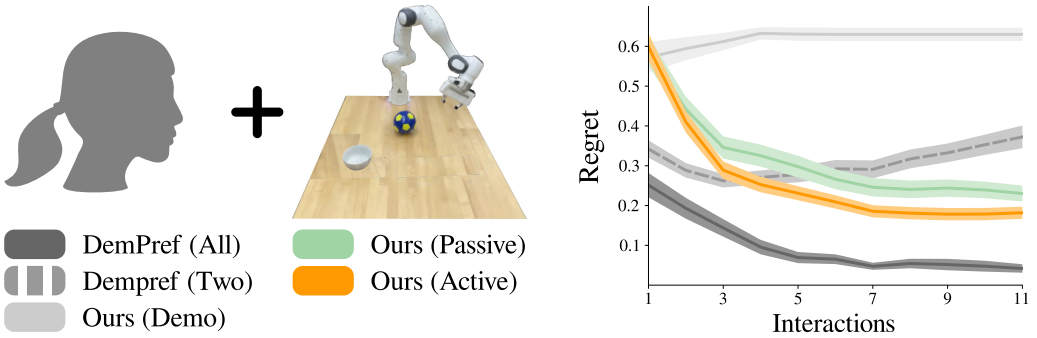

Fig. 8. Experimental results for simulated humans paired with a Franka Emika robot arm. (Left) We compare our approach to **DemPref** [4], a method for learning from demonstrations and preferences that assumes the robot has access to all task-related features. (Right) 500 humans attempt to teach the robot their desired task: Each user provides one demonstration followed by 10 preferences. In **DemPref (All)** the robot knows all three task-related features, in **DemPref (Two)** the robot is missing one feature, and in **Ours (Demo)** the robot only observes a single demonstration. We compare these baselines to our approach when the robot chooses preference queries at random, **Ours (Passive)**, and when the robot asks questions to gain as much information as possible, **Ours (Active)**. The shaded region is the standard error. **Our** approach outperforms **DemPref** when a feature is missing and the robot must learn an unexpected task.

proposed approach the robot can collect human preferences passively or actively. We will therefore consider two different versions of **Ours**: one where the robot gets the human's preferences from randomly chosen trajectories, **Ours (Passive)**, and one where the robot applies Equations (10) and (11) to select preference queries that will maximize the information the robot gains about θ, **Ours (Active)**. To make the comparison as fair as possible, we have given both **DemPref** and **Ours** the same dataset of 1000 trajectories from which to choose their preference queries. Each query in this dataset was sampled by choosing a random goal in the robot's workspace followed by generating two noisy trajectories to the goal. Finally, to show that obtaining human preferences improves the performance of our approach, we also include **Ours (Demo)**, a baseline where the robot learns from only one demonstration (without ever considering the human's preferences).

Procedure. We perform this experiment in a controlled environment by pairing simulated humans with a 7-DoF Franka Emika robot arm (Figure 8). The environment has three features: the distance of the robot from the bowl, the height of the robot from the table, and the distance between the robot and the ball. For each learning algorithm, we simulated 500 humans with randomly selected reward functions that depend on these three features. Put another way, every simulated human assigns a different relative importance to the task features following Equation (1). The robot does not know the human's reward function *a priori*. Over repeated interactions, the robot attempts to identify the correct reward (and the corresponding optimal trajectory) from the demonstrations and preferences of the current human user. Each simulated user selects their inputs to noisily optimize their internal reward function.

This experiment is designed similarly to the simulation in Section 7.1. We want to explore how our approach compares to **DemPref** in situations where the robot encounters a familiar task and settings where the robot is faced with new or unexpected tasks. We therefore compare **Ours** to **DemPref** (a) when all the features are known and (b) when one feature is missing. Recall that the task has three potential features. For case where one feature is missing, we performed separate trials where we removed either the first feature, the second feature, or the third feature; we then report the average across these runs. **Our** approach was never given any information about the features (i.e., **Ours** had no prior information about the task).

Dependent Variables. The simulated human worked with the real robot over 11 interactions. During the first interaction the human provides a demonstration, and during the next 10 interactions the robot collects preferences. After each interaction, the robot solves for its best guess of the task trajectory: We measure the performance of the robot learner using *regret* as defined in 14.

Hypothesis. For this experiment, we had the following hypotheses:

> **H8:** *Our method will outperform* ***DemPref*** *when the robot does not know all the task-related features.*
> **H9:** *Our method will converge to the correct trajectory more rapidly when choosing active preference queries as compared to passively collecting preferences or ignoring preferences altogether.*

Results. We summarize the results from this simulation in Figure 8. Remember that lower regret scores are better: After testing for the normality of data using Q-Q plots, a repeated measures ANOVA with Greenhouse–Geisser correction revealed that the learning algorithm had a significant main effect on regret by the end of the interactions ($F(1.994, 1996) = 366.897, p < .05$).

In settings where the robot encounters an expected task—i.e., when all the features of the environment are pre-programmed into the robot arm— **DemPref (All)** outperforms our proposed approach; **Ours (Passive)**: $p < .05$ and **Ours (Active)**: $p < .05$. However, when the robot is missing a task-related feature **DemPref (Two)** becomes confused by the human's feedback. Looking at Figure 8, we notice that **DemPref (Two)**'s performance decreases as the the human provides additional preferences: This occurs because the robot is misinterpreting the human's inputs as feedback about the two known features (instead of the one missing feature). By the end of the physical interactions, our proposed approach better understands the unexpected task than the baseline; **Ours (Passive)**: $p < .05$ and **Ours (Active)**: $p < .05$. Overall, the results here agree with hypothesis **H8**. From this result, we conclude that when the robot has access to the environment features, Bayesian inference approaches perform on par with or better than our proposed reward learning approach. However, in more open-ended cases where information on reward functions is missing, **Ours** is more suitable for robot learning.

We next compared different versions of our learning algorithm. First, we find that learning from both demonstrations and preferences provides more information about the task than learning only from the human's initial demonstrations. Post hoc tests show that **Ours (Demo)** has significantly higher regret than **Ours (Passive)** ($p < .05$) and **Ours (Active)** ($p < .05$). Next, we tested to see whether actively selecting preference queries would lead to faster adaptation than passive human feedback. Remember that for **Ours (Active)** the robot asked questions using Equation (10), while for **Ours (Passive)** the robot chose preference queries at random. We observe that **Ours (Active)** has significantly better performance than **Ours (Passive)** across 500 users ($p < 0.05$). We conclude that hypothesis **H9** is supported when robots learn from demonstrations and preferences.

8 CONCLUSION

In this article, we developed an alternate formalism for learning from physical human–robot interaction that unifies demonstrations, corrections, and preferences. When humans and robots share spaces, physical interaction is inevitable. Robots should leverage this interaction to learn from the human and improve their own behavior. But physical interaction takes many forms: Humans can kinesthetically guide the robot throughout an entire task, fine-tune snippets of the robot's motion, or indicate which robot trajectories they prefer. Existing methods either learn from one of these interaction modalities, or combine multiple modalities by assuming prior information about the human's task. Instead, we introduce an end-to-end framework that (a) learns a reward function from scratch and then (b) optimizes this reward function to obtain robot trajectories.

Our key technical insight was that we can unite demonstrations, corrections, and preferences within the same framework by learning to assign higher rewards to these human inputs than to nearby alternatives. We first described a way to generate trajectory modifications. Next, we derived loss functions for learning from demonstrations, corrections, and preferences, and used these loss functions to train an ensemble of reward models. We also enabled robots to actively prompt the human and gain information about the correct task behavior. Finally, we converted the robot's learned rewards to robot trajectories using constrained optimization. Our framework was specifically developed for robot arms performing manipulation tasks: Through simulations and a user study, we compared our approach to multiple state-of-the-art baselines. Our results indicate that—when the robot knows what tasks to expect—our learning approach is comparable to existing methods that rely on pre-programmed features. However, when the robot encounters unexpected tasks (or when the robot must learn a new task from scratch), our method outperforms the interactive reward learning and imitation learning baselines.

Limitations and Future Works. Our work is a step towards seamless communication between humans and robot arms. Because our system can learn new behaviors, one practical concern is *safety*: We must ensure that the robot learns trajectories that are safe for shared human–robot spaces. For instance, in our running example, the robot arm should never learn to swing towards the human or run into the table. If the designer knows these safety constraints *a priori* we can embed them within our approach. Specifically, designers could augment Equation (13) to constrain the robot to have a certain workspace or speed thresholds. However, if designers do not impose any limits, we currently cannot guarantee that our robot will learn human-friendly behaviors.

One assumption throughout our work is that the human's inputs are noisily optimal. We assume that—when the human makes a correction or provides a preference—on average their input is better aligned with their underlying reward function than the alternatives. However, in some settings and modalities, the human's inputs may have a persistent bias, leading to suboptimal demonstrations, corrections, or preferences. Imagine a person teaching a robot to move across the table: If this human teacher cannot reach the opposite side of the table, all of their demonstrations may only move the robot part of the way to their intended goal. Existing works have explored methods for extrapolating from suboptimal demonstrations to reach performance that exceeds the given feedback [8, 9, 43]. We hypothesize that these methods could be combined with our reward learning approach by leveraging the diverse types of human feedback. For example, the user may have a persistent bias in their demonstrations but not their preferences (e.g., in our example, the human cannot reach across the table but they can compare two trajectories that do so). Hence, we envision an iterative solution where the robot uses our approach to build a reward model from different types of feedback while identifying which of these modalities are biased and which are reliable.

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

[2] Baris Akgun, Maya Cakmak, Karl Jiang, and Andrea L. Thomaz. 2012. Keyframe-based learning from demonstration. *International Journal of Social Robotics* 4, 4 (2012), 343–355.

[3] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57, 5 (2009), 469–483.

[4] Erdem Bıyık, Dylan P. Losey, Malayandi Palan, Nicholas C. Landolfi, Gleb Shevchuk, and Dorsa Sadigh. 2022. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research* 41, 1 (2022), 45–67.

[5] Andreea Bobu, Andrea Bajcsy, Jaime F. Fisac, Sampada Deglurkar, and Anca D. Dragan. 2020. Quantifying hypothesis space misspecification in learning from human–robot demonstrations and physical corrections. *IEEE Transactions on Robotics* 36, 3 (2020), 835–854.

[6] Andreea Bobu, Marius Wiggert, Claire Tomlin, and Anca D. Dragan. 2022. Inducing structure in reward learning by learning features. *The International Journal of Robotics Research* 41, 5 (2022), 497–518.
[7] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. 2019. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *Proceedings of the International Conference on Machine Learning*. 783–792.
[8] Daniel S. Brown, Wonjoon Goo, and Scott Niekum. 2020. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Proceedings of the Conference on Robot Learning*. PMLR, 330–359.
[9] Letian Chen, Rohan Paleja, and Matthew Gombolay. 2021. Learning from suboptimal demonstration via self-supervised reward regression. In *Proceedings of the Conference on Robot Learning*. 1262–1277.
[10] Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.
[11] Thomas M. Cover. 1999. *Elements of Information Theory*. John Wiley & Sons.
[12] Agostino De Santis, Bruno Siciliano, Alessandro De Luca, and Antonio Bicchi. 2008. An atlas of physical human–robot interaction. *Mechanism and Machine Theory* 43, 3 (2008), 253–270.
[13] Anca D. Dragan, Katharina Muelling, J. Andrew Bagnell, and Siddhartha S. Srinivasa. 2015. Movement primitives via optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 2339–2346.
[14] Justin Fu, Katie Luo, and Sergey Levine. 2018. Learning robust rewards with adversarial inverse reinforcement learning. In *Proceedings of the International Conference on Learning Representations*.
[15] Philip E. Gill, Walter Murray, and Michael A. Saunders. 2005. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review* 47, 1 (2005), 99–131.
[16] Sami Haddadin, Alin Albu-Schaffer, Alessandro De Luca, and Gerd Hirzinger. 2008. Collision detection and reaction: A contribution to safe physical human–robot interaction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 3356–3363.
[17] Sami Haddadin and Elizabeth Croft. 2016. Physical Human–Robot Interaction. In *Springer Handbook of Robotics*, Bruno Siciliano and Oussama Khatib (Eds.). Springer International Publishing, Cham, 1835–1874.
[18] Michael Hagenow, Emmanuel Senft, Robert Radwin, Michael Gleicher, Bilge Mutlu, and Michael Zinn. 2021. Corrective shared autonomy for addressing task variability. *IEEE Robotics and Automation Letters* 6, 2 (2021), 3720–3727.
[19] Neville Hogan. 1984. Impedance control: An approach to manipulation. In *Proceedings of the American Control Conference*. 304–313.
[20] Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S. Brown, and Ken Goldberg. 2021. ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning. In *Proceedings of the Conference on Robot Learning*.
[21] Taylor A. Howell, Brian E. Jackson, and Zachary Manchester. 2019. ALTRO: A fast solver for constrained trajectory optimization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 7674–7679.
[22] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. 2018. Reward learning from human preferences and demonstrations in Atari. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*.
[23] Ashesh Jain, Shikhar Sharma, Thorsten Joachims, and Ashutosh Saxena. 2015. Learning preferences for manipulation tasks from online coactive feedback. *The International Journal of Robotics Research* 34, 10 (2015), 1296–1313.
[24] Hong Jun Jeon, Smitha Milli, and Anca Dragan. 2020. Reward-rational (implicit) choice: A unifying formalism for reward learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. 4415–4426.
[25] Mrinal Kalakrishnan, Peter Pastor, Ludovic Righetti, and Stefan Schaal. 2013. Learning objective functions for manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 1331–1336.
[26] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. 2019. HG-DAgger: Interactive imitation learning with human experts. In *Proceedings of the International Conference on Robotics and Automation*. 8077–8083.
[27] Mahdi Khoramshahi and Aude Billard. 2019. A dynamical system approach to task-adaptation in physical human–robot interaction. *Autonomous Robots* 43, 4 (2019), 927–946.
[28] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
[29] Kimin Lee, Laura M. Smith, and Pieter Abbeel. 2021. PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *Proceedings of the International Conference on Machine Learning*. 6152–6163.
[30] Mengxi Li, Alper Canberk, Dylan P. Losey, and Dorsa Sadigh. 2021. Learning human objectives from sequences of physical corrections. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 2877–2883.

[31] Yanan Li, Gerolamo Carboni, Franck Gonzalez, Domenico Campolo, and Etienne Burdet. 2019. Differential game theory for versatile physical human–robot interaction. *Nature Machine Intelligence* 1, 1 (2019), 36–43.
[32] Dylan P. Losey, Andrea Bajcsy, Marcia K O'Malley, and Anca D. Dragan. 2022. Physical interaction as communication: Learning robot objectives online from human corrections. *The International Journal of Robotics Research* 41, 1 (2022), 20–44.
[33] Dylan P. Losey, Craig G. McDonald, Edoardo Battaglia, and Marcia K. O'Malley. 2018. A review of intent detection, arbitration, and communication aspects of shared control for physical human.robot interaction. *Applied Mechanics Reviews* 70, 1 (2018), 010804.
[34] Dylan P. Losey and Marcia K. O'Malley. 2017. Trajectory deformations from physical human–robot interaction. *IEEE Transactions on Robotics* 34, 1 (2017), 126–138.
[35] Dylan P. Losey and Marcia K. O'Malley. 2019. Learning the correct robot trajectory in real-time from physical human interactions. *ACM Transactions on Human-Robot Interaction* 9, 1 (2019), 1–19.
[36] Christopher Lucas, Thomas Griffiths, Fei Xu, and Christine Fawcett. 2008. A rational model of preference learning and choice prediction by children. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*.
[37] R. Duncan Luce. 2012. *Individual Choice Behavior: A Theoretical Analysis*. Courier Corporation.
[38] Alexander Mörtl, Martin Lawitzky, Ayse Kucukyilmaz, Metin Sezgin, Cagatay Basdogan, and Sandra Hirche. 2012. The role of roles: Physical cooperation between humans and robots. *The International Journal of Robotics Research* 31, 13 (2012), 1656–1674.
[39] Selma Musić and Sandra Hirche. 2017. Control sharing in human-robot team interaction. *Annual Reviews in Control* 44, (2017), 342–354.
[40] Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.
[41] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, Jan Peters, and others. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics* 7, 1–2 (2018), 1–179.
[42] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 627–635.
[43] Mariah L. Schrum, Erin Hedlund-Botti, Nina Moorman, and Matthew C. Gombolay. 2022. MIND MELD: Personalized meta-learning for robot-centric imitation learning. In *Proceedings of the 2022 17th ACM/IEEE International Conference on Human-Robot Interaction*. IEEE, 157–165.
[44] Mariah L. Schrum, Michael Johnson, Muyleng Ghuy, and Matthew C. Gombolay. 2020. Four years in review: Statistical practices of likert scales in human–robot interaction studies. In *Companion of the 2020 ACM/IEEE International Conference on Human–Robot Interaction*. 43–52.
[45] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. 2014. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research* 33, 9 (2014), 1251–1270.
[46] Roger N. Shepard. 1957. Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika* 22, 4 (1957), 325–345.
[47] Jonathan Spencer, Sanjiban Choudhury, Matthew Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Sidd Srinivasa. 2022. Expert intervention learning. *Autonomous Robots* 46, 1 (2022), 99–113.
[48] Hang Yin, Francisco S. Melo, Ana Paiva, and Aude Billard. 2019. An ensemble inverse optimal control approach for robotic task learning and adaptation. *Autonomous Robots* 43, 4 (2019), 875–896.
[49] Songyuan Zhang, Zhangjie Cao, Dorsa Sadigh, and Yanan Sui. 2021. Confidence-aware imitation learning from demonstrations with varying optimality. In *Proceedings of the Conference on Neural Information Processing Systems*.
[50] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence*. 1433–1438.

Received 14 June 2022; revised 24 May 2023; accepted 11 August 2023