StROL: Stabilized and Robust Online Learning from Humans

Shaunak A. Mehta, Forrest Meng, Andrea Bajcsy, and Dylan P. Losey

***Abstract*— Today's robots can learn the human's reward function online, during the current interaction. This real-time learning requires fast but approximate learning rules; when the human's behavior is noisy or suboptimal, today's approximations can result in unstable robot learning. Accordingly, in this paper we seek to enhance the robustness and convergence properties of gradient descent learning rules when inferring the human's reward parameters. We model the robot's learning algorithm as a *dynamical system* over the human preference parameters, where the human's true (but unknown) preferences are the equilibrium point. This enables us to perform Lyapunov stability analysis to derive the conditions under which the robot's learning dynamics converge. Our proposed algorithm (StROL) takes advantage of these stability conditions offline to modify the original learning dynamics: we introduce a corrective term that expands the basins of attraction around likely human rewards. In practice, our modified learning rule can correctly infer what the human is trying to convey, even when the human is noisy, biased, and suboptimal. Across simulations and a user study we find that StROL results in a more accurate estimate and less regret than state-of-the-art approaches for online reward learning. See videos here: https://youtu.be/uDGpkvJnY8g**

I. Introduction

Robots can learn in real-time from human feedback. Consider Figure 1, where a robot arm is carrying a cup of water close to a pitcher. Existing work enables this robot to learn the human's preferences (i.e., their desired behavior) online based on the human's actions. For instance, if a human pushes the arm away from the pitcher, the robot will modify its current trajectory to keep cups farther from pitchers.

To achieve this real-time performance, today's robots often use fast but approximate learning rules. Methods like [1]–[7] maintain a point estimate over the human's preferences, and update this estimate online using gradient descent. This works efficiently when user's inputs are perfectly aligned with the assumptions of the robot's learning algorithm. But when the human inevitably deviates, today's fast but approximate learning rules are highly sensitive: noisy, biased, and suboptimal humans can lead to *unstable* robot learning [3]. Returning to Figure 1, a human that overcorrects the arm causes the system to oscillate between avoiding and approaching the pitcher, continually interacting without ever converging to the human's true preference.

In this paper we tackle the question:

How can robots leverage online learning algorithms while ensuring robustness with suboptimal human feedback?

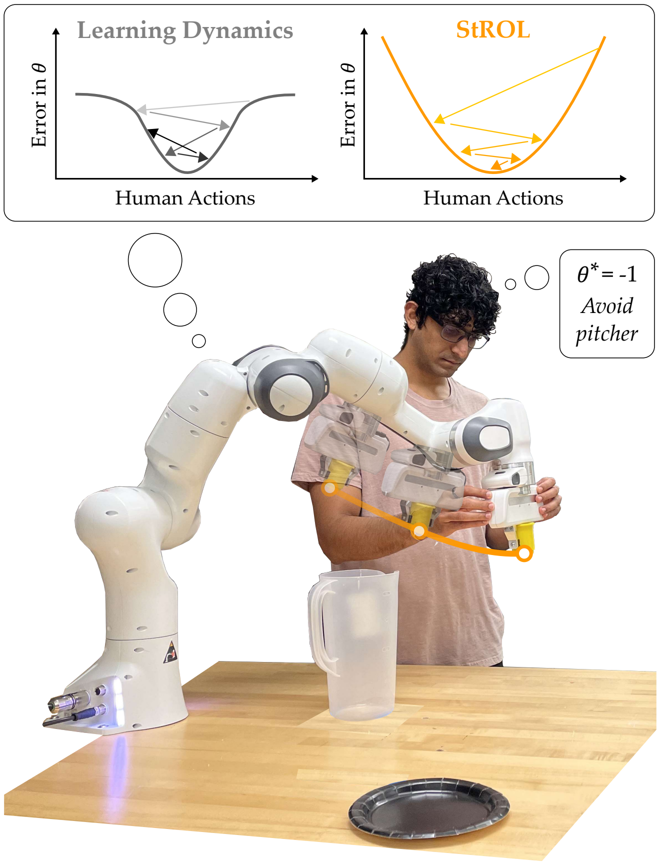

Fig. 1. Human physically correcting a robot arm to convey their reward parameters θ^*. The robot learns online, and updates its point estimate θ after each human action. (Left) When the human takes noisy or suboptimal actions, the given learning dynamics can become unstable and fail to converge to θ^*. (Right) We modify these dynamics to expand the basins of attraction and increase robustness in the face of suboptimal humans.

To answer this question we will model the robot's learning algorithm as a *dynamical system* in the continuous space of preference parameters. By treating online learning as a dynamical system we enable a *control theoretic* perspective that formally studies the desired convergence and robustness properties of the robot learner.

Our key idea is to augment the initial learning dynamics with an added corrective term. The point of this corrective term is to expand the basin of attraction of the robot's learning dynamics: the augmented system should now converge to the human's true preferences under a larger set of initial conditions and suboptimal human inputs. But how do we find the optimal values for this corrective term? Our work recognizes that not all preferences are created equal: even though the preference space is inherently continuous, there

S.A. Mehta, F. Meng, and D.P. Losey are with the Collaborative Robotics Lab (Collab), Dept. of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24061. A. Bajcsy is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15289.
Corresponding author's email: mehtashaunak@vt.edu

are are only a few relevant modes that the human might want. Consider Figure 1: along the spectrum of distances to pitcher, either (a) the human does not care if the robot moves over the pitcher or (b) the human wants the robot to avoid this space. We model these modes via multimodal priors over the human's preferences. Combining these priors with Lyapunov stability analysis, we introduce an offline learning procedure for optimizing the corrective term. In practice, our modified learning rule shapes the robot's basin of attraction around each functionally different preference, enabling fast convergence despite suboptimal human feedback. Returning to Figure 1, under our approach the human can provide unintended forces — e.g., accidentally push too hard — and still convey their preference for keeping away from pitchers.

Overall, we make the following contributions:

Formulating Conditions for Convergence. We write real-time learning from human feedback as a dynamical system where the human's true preferences are the equilibrium point. We then apply Lyapunov stability analysis to derive the conditions for converging to this equilibrium.

Learning to Learn from Suboptimal Humans. We change the learning dynamics of the system to include an additional term. Offline we train this term to shape the dynamics and increase the basin of attraction around preferences sampled from a known prior. We refer to the resulting algorithm as **StROL**: **St**abilized and **R**obust **O**nline **L**earning.

Collaborating with Imperfect Users. We perform simulations and a user study across scenarios with robot arms and autonomous driving. We demonstrate that our proposed learning dynamics are more robust to noisy and suboptimal humans than state-of-the-art alternatives.

II. Related Works

We focus on real-time learning from humans. We seek to learn what the human wants (i.e., preferences) while framing learning in human-robot interaction as a dynamical system.

Online Robot Learning from Humans. Online reward learning explores how robots can infer preferences from nearby humans during an interaction [8]. In settings where humans correct the robot's behavior — or intervene when the robot makes a mistake — online learning enables rapid robot adaptation. Prior works have applied online learning from human feedback to autonomous vehicles [9], assistive exoskeletons [10], and robot arms [11]. To enable real-time performance, online learning often requires simplifying assumptions. For instance, relevant works like [1]–[7] maintain a point estimate of what the human wants, and update this estimate using gradient descent. Unfortunately, the approximations needed for online learning also make the system sensitive to suboptimal human inputs. When the user inevitably messes up (and incorrectly intervenes) the robot may learn the wrong preferences [3] or misrepresent the human's true intentions [5]. Instead of thinking of this as a *learning* problem, we instead treat this as a *control* problem: how should robots modify their learning rule to ensure effective performance across suboptimal human inputs?

Learning from Humans as a Dynamical System. As a step towards fast and seamless adaptation, we will model online robot learning from humans as a *dynamical system*. Recent works have found different ways to incorporate learning mechanisms into the dynamics models of human-robot interaction. This includes shared control settings where the robot adjusts its desired trajectory based on applied forces and torques [12]–[14], jointly learning a model of the human policy and physical dynamics [15], modeling the human's learning process as a dynamical system [16], and dynamic movement primitives that react to human motions or demonstrations [17]. Across many of these previous works, the authors apply control theory to demonstrate that their proposed dynamical system is stable. But we take the opposite perspective: instead of using dynamics and control to *reactively analyze* a given learning method, we will employ control theory to *actively shape* the robot's learning rule in ways that lead to robust convergence.

Priors over Human Preferences. A core idea for our approach is that — even when the robot is learning in continuous spaces — there are distinct *modes* of human preferences. Think back to our motivating example: there is a continuous spectrum of distances between the robot and pitcher that the human could prefer. However, at a high level, the space of preferences can be divided into two distinct preference modes: avoiding the pitcher or ignoring it altogether. Research in cognitive science and machine learning suggests that humans have *strong priors* over how other people with act [18], [19] and what sort of behaviors are reasonable [20], [21]. Robots can often obtain these priors from data: recent works have shown that large language models accurately predict the different actions a human might take [22]. Building on these works, we leverage intuitive, multimodal priors over the continuous preference space to shape the robot's learning rule, yielding more robust and efficient learning from human data.

III. Problem Statement

We consider interactive scenarios where robots learn from humans in *real-time*. This includes settings where the robot performs a task and the human is purely a teacher (e.g., a human physically correcting a robot arm), or settings where the human and robot are both performing a task in the same environment (e.g., an autonomous car driving near a pedestrian). In both settings the human has a task that they want to perform, and the robot is trying to learn this task from the human's actions. In this section we formulate real-time human-robot interaction as a dynamical system with two parts: *state dynamics* and robot's *learning dynamics*. We assume the state dynamics are known (i.e., the robot has a model of the environment), and the robot is initialized with some learning dynamics (i.e., the designer provides a baseline learning rule). We will explore how to modify these initial learning dynamics to stabilize interactions with humans that take noisy, imperfect, or unexpected actions.

Physical Dynamics. Let $x \in \mathcal{X}$ denote the system state. In our experiments x can be the joint position of a robot arm, or the combined pose of an autonomous car and human pedestrian. At each timestep t the human takes action $u_{\mathcal{H}} \in \mathcal{U}_{\mathcal{H}}$ and the robot takes action $u_{\mathcal{R}} \in \mathcal{U}_{\mathcal{R}}$. The system state transitions according to the known, deterministic, discrete-time *state dynamics*:

$$x^{t+1} = f(x^t, u^t_{\mathcal{H}}, u^t_{\mathcal{R}}) \tag{1}$$

The interaction ends after a total of T timesteps. We emphasize that the human and robot only collaborate for a *single* interaction; the robot *does not* repeatedly work with the same human across multiple, separate interactions.

Unknown Parameters. During interaction the robot optimizes its reward function. There may be some aspects of this reward that the robot already knows — e.g., the robot arm should carry water across the table. However, there are also parameters the robot does not know — like whether the robot should avoid moving over the pitcher. Let the true objective be $R(x, \theta^*) \rightarrow \mathbb{R}$, where $\theta^* \in \mathbb{R}^d$ is a d-dimensional vector of *correct* reward parameters (e.g., the task that the robot should optimize for). Returning to our motivating examples, θ^* could capture how the robot arm should carry a glass, or where and when the pedestrian will cross the road. The robot does not know θ^* and must learn these parameters from the human data–specifically, observations of the human's actions.

Prior. Although the robot does not know θ^* a priori, we assume the robot does have a prior $P(\theta)$ over the continuous space of reward parameters. This prior encodes what types of behaviors the person might want: returning to Figure 1, the prior could be a bimodal distribution signifying that the human either wants to avoid the pitcher or does not care about moving over this pitcher. In our experiments we set $P(\theta)$ as a *multimodal distribution*, where each mode corresponds to a different type of behavior.

Learning Dynamics. The robot is trying to learn the true reward parameters θ^*. For tractable, real-time learning, the robot maintains a *point estimate* of these true reward parameters: this point estimate is the robot's best guess of θ^*. Let θ^t denote the robot's point estimate at timestep t, and remember that $\theta \in \Theta$ lies in a continuous Euclidean space.

Building on the state-of-the-art in online learning from human feedback [2]–[6], [23], we use gradient ascent to capture the deterministic dynamics of the point estimate:

$$\theta^{t+1} = \theta^t + \alpha \cdot g(x^t, u^t_{\mathcal{H}}, u^t_{\mathcal{R}}, \theta^t) \tag{2}$$

Here $\alpha \geq 0$ is the learning rate and $g(x, u_{\mathcal{H}}, u_{\mathcal{R}}, \theta) \rightarrow \mathbb{R}^d$ is a function that determines how the point estimate changes in response to human action $u_{\mathcal{H}}$. Throughout the paper, we use the term *learning dynamics* to refer to Equation (2) and g interchangeably. The choice of g is up to the designer; in our analysis, our only requirement is that g in Equation (2) must depend on human action $u_{\mathcal{H}}$.

Example. Below we list one common choice of learning rule. Let $x_{\mathcal{H}} = f(x, u_{\mathcal{H}}, u_{\mathcal{R}})$ be the next state if the human takes action $u_{\mathcal{H}}$, and let $x_{\mathcal{R}} = f(x, 0, u_{\mathcal{R}})$ be the next state if only the robot acts. Related works [2]–[5] update the point estimate to increase the reward for the human's corrected state $x_{\mathcal{H}}$ as compared to the default state $x_{\mathcal{R}}$:

$$g = \nabla_\theta \big(R(x_{\mathcal{H}}, \theta) - R(x_{\mathcal{R}}, \theta)\big) \tag{3}$$

We will use Equation (3) in our experiments. However, our underlying method is not tied to this specific instantiation.

Perturbations. We have formulated human-robot interaction as a dynamical system with state dynamics in Equation (1) and learning dynamics in Equation (2). Ideally, we want the estimate θ to converge towards the human's preferences θ^* so that the robot can optimize the correct reward function. This would be straightforward if the human's inputs $u_{\mathcal{H}}$ exactly aligned with the robot's learning algorithm. Consider our motivating example of a human teaching a robot arm how to carry a cup: if the human physically corrects the robot such that $g(u_{\mathcal{H}})$ causes $\theta^{t+1} \rightarrow \theta^*$, then the robot will learn the correct task. But what if the human is not a perfect teacher? We recognize that humans are *suboptimal* agents [24], [25], and thus our dynamical system must be *robust* to perturbations in the human's actions. We want θ to converge to the unknown equilibrium θ^* even when the human's inputs are not precisely aligned with the given dynamical system.

IV. Shaping the Learning Dynamics to Enlarge Basins of Attraction

In this section we present our control theoretic approach to modify the learning dynamics by adding a corrective term, making the system robust to suboptimal humans. Our proposed method is based on stabilizing the learning dynamics around a preference equilibrium $\theta = \theta^*$. More specifically, we leverage Lyapunov stability analysis in Section IV-A to derive a condition which guarantees that the error between θ and θ^* is asymptotically decreasing. This condition gives us a stable region of actions that the human can take to drive the robot's point estimate θ towards the human's true preference parameter θ^*. Next, in Section IV-B we focus on learning the optimal values for the correction term *offline*. Leveraging the Lyapunov stability condition in addition to our model of the human and priors over their preferences, we modify g to expand the basins of attraction so that the learning dynamics align with the human's behavior.

A. Deriving a Stability Condition

We know that human teachers will not always provide perfect, consistent inputs. Rather than assuming the human selects a single optimal choice of $u_{\mathcal{H}}$ to teach the robot, we are instead interested in the *domain* of human actions that convey θ^*. Put another way, under what conditions does the human's action $u_{\mathcal{H}}$ cause the estimate θ to converge to θ^*?

Recall that our key idea is to augment the initial learning dynamics of the system by adding a correction term. Thus, we write the updated learning rule as:

$$\tilde{g}^t = g^t + \hat{g}^t \tag{4}$$

where g^t is short for the original rule $g(x^t, u^t_{\mathcal{H}}, u^t_{\mathcal{R}}, \theta^t)$, and $\hat{g}$ denotes the correction term $\hat{g}(x^t, u^t_{\mathcal{H}}, u^t_{\mathcal{R}}, \theta^t)$. By introducing this correction term in the learning dynamics, we aim to expand the basins of attraction and stabilize

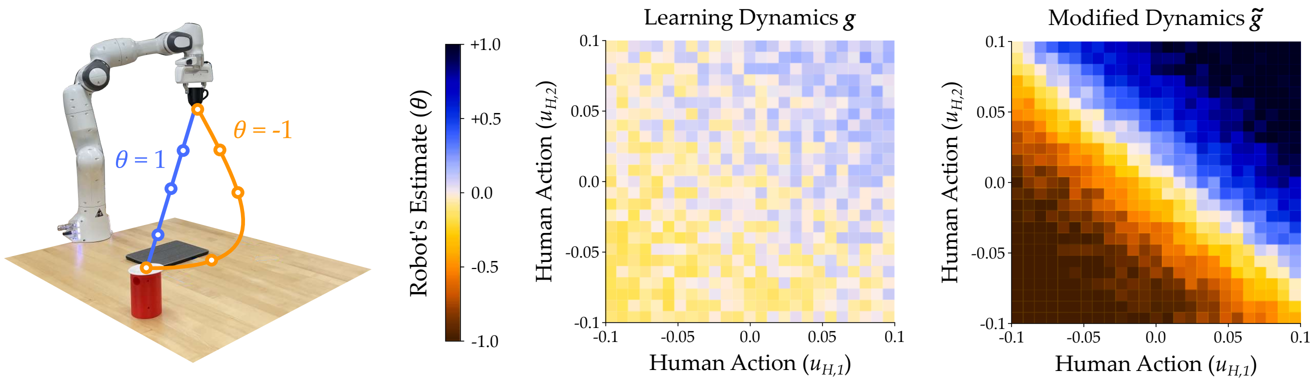

Fig. 2. Example of how our proposed dynamics approach expands the learning basin of attraction. (Left) The robot is carrying a cup across the table. The robot does not know how it should move near a laptop: when $\theta = +1$ the human wants the robot to move straight to the goal, and when $\theta = -1$ the human wants the robot to avoid moving above the laptop. (Right) Plots of the robot's estimate θ as a function of the human's action $u_{\mathcal{H}}$ at the start state. With the original learning dynamics g the learning is inconsistent and gradual (i.e., nearby actions can convey either ignoring or avoiding the laptop). But the modified learning dynamics $\tilde{g}$ expands the basin of attraction, so that nearby actions teach the robot the same parameters.

the dynamics across a wider range of human inputs $u_{\mathcal{H}}$. Substituting this modified learning rule into Equation (2), we get the updated learning dynamics for the system:

$$\theta^{t+1} = \theta^t + \alpha \cdot (g^t + \hat{g}^t) \tag{5}$$

Ideally, our system learns to drive the robot's current estimate of the human parameters θ^t to the equilibrium θ^* based on the human's actions. Define $e^t = \theta^* - \theta^t$ as the error in the robot's estimate of the human preferences at the current timestep. To identify the set of human actions that drives $\theta^t \rightarrow \theta^*$ and causes the system to converge to equilibrium, we will apply Lyapunov stability analysis.

Let the Lyapunov candidate function be $V^t = \|e^t\|_2^2$. Note that this function is positive definite and radially unbounded, i.e., the function cannot be 0 at any point except for the equilibrium ($\theta^t = \theta^*$) and $V^t \rightarrow \infty$ as $e_t \rightarrow \infty$. The time derivative of the candidate function is:

$$\dot{V} \approxeq V^{t+1} - V^t = \|e^{t+1}\|_2^2 - \|e^t\|_2^2 \tag{6}$$

For global asymptotic stability of the system around the equilibrium, according to Lyapunov's Direct Method we need that $\dot{V} < 0$ [26]. Substituting this condition into Equation (6), the sufficient condition for convergence becomes $\|e^{t+1}\|^2 < \|e^t\|^2$. Plugging in e^t and the modified learning dynamics from Equation (5), we reach:

$$\|\theta^* - \theta^t - \alpha \cdot \tilde{g}^t\|_2^2 < \|\theta^* - \theta^t\|_2^2 \tag{7}$$

Expanding this inequality and rearranging the terms, the sufficient condition for global asymptotic stability is:

$$\alpha^2 \|\tilde{g}^t\|_2^2 - 2\alpha(e^t \cdot \tilde{g}^t) < 0 \tag{8}$$

Any action $u_{\mathcal{H}}$ that satisfies this constraint lies in the basin of attraction and will eventually drive $\theta^t \rightarrow \theta^*$ (i.e. the set of these actions define a *stable region* of human inputs). Conversely, any human actions $u_{\mathcal{H}}$ that does not satisfy this constraint will cause the error in θ to remain constant or increase (i.e. an unstable set of human inputs). We emphasize that the condition derived in Equation (8) depends on how g maps the human's actions to changes in θ: a given human action may satisfy Equation (8) for one choice of learning dynamics g but not for another. We also note that a more negative value in this constraint suggests that the human actions $u_{\mathcal{H}}$ are causing θ to converge more rapidly.

B. *StROL: Learning the Correction Term*

The Lyapunov stability condition provides a functional requirement for robustness of the overall learning dynamics, $\tilde{g}$. Although we introduced a modification to the learning dynamics via the addition of $\hat{g}$, how do we find the correct values for this term? Here we focus on finding the optimal values for $\hat{g}$. Notice that to leverage the convergence condition defined in Equation (8), we need some prior information about the human preferences θ^* (to evaluate e^t) and the range of actions $u_{\mathcal{H}}$ the human may take (to evaluate $\tilde{g}$).

Prior. In any given environment there are a few different tasks that a human is likely to perform, i.e., we assume a prior over the space of human preferences $P(\theta)$. This prior is a designer specified parameter that should capture the different types of possible human behaviors in a given environment. In our experiments, we select $P(\theta)$ as a multimodal normal distribution. To calculate e^t, we sample $\theta^* \sim P(\theta)$.

Human Model. To get the human actions $u_{\mathcal{H}}$ for evolving the learning dynamics $\tilde{g}$, we need a model of the suboptimal human. Let us first consider an optimal human. We recognize that an optimal human trying to convey their preference parameters to the robot will always take actions $u_{\mathcal{H}}^*$ that drive the robot's estimate $\theta^t \rightarrow \theta^*$. Offline, we generate these optimal actions $u_{\mathcal{H}}^*$ by simulating a human whose preference parameters θ^* are sampled from $P(\theta)$:

$$u_H^{*\,t} = \min_{u_H \in U} \theta^* - (\theta^t + \alpha g^t), \theta^* \sim P(\theta) \tag{9}$$

In practice the human is imperfect and will not always take optimal actions. Without loss of generality, we write the actions of a suboptimal human as $u_{\mathcal{H}} = u_{\mathcal{H}}^* + \delta$, where δ represents the noise, bias or any other factor that perturbs the human. The choice of δ is up to the designer and depends on their model of the human and environment. For example, in our experiments we set $\delta \sim \mathcal{N}(\epsilon, \sigma)$, where σ is the variance from the optimal actions and ϵ is a consistent bias. Note that the more information the designer has about the human,

Algorithm 1 StROL: Stabilized and Robust Online Learning

```
Define original learning dynamics $g$ ▷ see Equation (3)
Randomly initialize corrective term $\hat{g}$
for $i = 1, 2, \cdots$ do
    Initialize the empty training dataset $\mathcal{D}$
    for $j = 1, 2, \cdots, N$ do
        Sample $(x, \theta, \theta^*)$ tuple, where $\theta^* \sim P(\theta)$
        Get optimal actions $u^*_{\mathcal{H}}$ using Equation (9)
        $u_{\mathcal{H}} \leftarrow u^*_{\mathcal{H}} + \delta$
        Update the training dataset $\mathcal{D} \leftarrow (x, u_{\mathcal{H}}, \theta^*, \theta)$
    end for
    Compute the loss $\mathcal{L}$ using Equation (10)
    Update $\hat{g}$ to minimize $\mathcal{L}$
end for
```

the more accurate they can make this model of the human's actions. This in turn will lead to a corrective term that is better suited to the current user.

Offline Learning. Equipped with the condition for convergence in Equation (8), the prior over the human preferences $\mathcal{P}(\theta)$, and a model of the suboptimal human's actions, we can now train $\hat{g}$ *offline* to increase the basin of attraction around the human preferences (see Algorithm 1). We model $\hat{g}$ as a neural network and leverage our stability condition as a loss function when training the network:

$$\mathcal{L} = \sum_{\theta^*, u_{\mathcal{H}} \in \mathcal{D}} \alpha^2 \|\tilde{g}^t\|_2^2 - 2\alpha(e^t \cdot \tilde{g}^t) \tag{10}$$

where the dataset $\mathcal{D}$ is generated by first sampling the human's preference parameters $\theta^* \sim P(\theta)$. For each of these sampled preference parameters, we then initialize the system in a random state $x \in \mathcal{X}$ and use Equation (9) to generate the optimal action $u^*_{\mathcal{H}}$. Finally, we perturb these optimal actions to get the suboptimal human actions $u_{\mathcal{H}}$ that we use to train the model in Equation (10).

Example. In our experiments $\hat{g}$ is a fully connected 5 layer multi-layer perception with a rectified linear unit activation function. The output of $\hat{g}$ is bounded by a $\tanh(\cdot)$ activation function such that $\|\hat{g}\| \leq \|g\|$. This prevents the correction term $\hat{g}$ from overpowering the original learning dynamics g. In Figure 2 we show an example of how our corrective term modifies the learning dynamics to expand the basin of attraction. We first trained $\hat{g}$ offline using our StROL algortihm (Algorithm 1). We next measured the estimate θ that the robot learns with either the original dynamics g or the modified learning dynamics $\tilde{g} = g + \hat{g}$. In this example $\hat{g}$ expands the basin of attraction so that one region of human actions teaches the robot to avoid the laptop ($\theta \rightarrow -1$), and the opposite region of human actions causes the robot to ignore the laptop ($\theta \rightarrow +1$).

V. Simulations

We have developed an approach to expand the basins of attraction when learning in real-time from suboptimal humans. Here we perform controlled simulations, and examine whether our learning and dynamics framework results in more robust performance as compared to state-of-the-art baselines. We consider two simulated environments: (a) a multi-agent driving scenario where the robot car needs to learn the human's driving style to avoid a collision, and (b) a household setting where the human physically corrects a robot arm. In both environments we simulate humans whose actions are sampled with increasing levels of noise and bias.

Independent Variables. We compare our proposed algorithm (**StROL**) to four baselines that update a point estimate θ using different versions of the gradient-based learning rule in Equation (2). Gradient descent (**Gradient**) directly uses Equation (2) with learning dynamics g. One-at-a-time (**One**) [3] modifies these learning dynamics to account for noisy and imprecise humans: instead of updating each element of θ at every timestep, the robot only changes the element of θ that best aligns with the human's action. Misspecified Objective Functions (**MOF**) [5] also modifies the learning dynamics in Equation (2) to accommodate unexpected human behaviors. Specifically, here the robot ignores — and does not learn from — human actions $u_{\mathcal{H}}$ that are not aligned with any of the parameters in θ. Finally, we test an ablation of our proposed approach that we refer to as End-to-End (**e2e**). In **StROL** the robot's learning dynamics $\tilde{g}$ are the sum of the original dynamics g and the corrective term $\hat{g}$. We hypothesize that g provides an important starting point (i.e., the designer's knowledge) about the correct learning dynamics. In **e2e** we test whether including g is really necessary by setting $\tilde{g} = \hat{g}$, and training the robot's learning rule completely from scratch. **e2e** uses the exact same network architecture for $\hat{g}$ as **StROL**.

Environments. We tested two settings: a multi-agent **Highway** environment and a collaborative **Robot** environment.

In **Highway** a robot car is driving in front of a human car on a two-lane highway. We simulate both vehicles in CARLO [27]. The cars start in the left lane with the human behind the autonomous car. Both the human and robot cars have two-dimensional action spaces. For this simulation, we consider three features, (a) *distance* between the human and robot cars, (b) *speed* of the robot car and (c) heading direction of the human car indicating whether or not the human will *change lane*. The robot's goal is to minimize the distance travelled and avoid any collisions. To train the corrective term $\hat{g}$ in **StROL** and **e2e** we assume a bimodal prior: either (a) the human car will change lanes and then pass the robot car (i.e. the human car does not care about *distance* but has a preference for speed and change lane), or (b) the human will follow the robot until the robot switches lanes (the human car does not want to *change lane* and maintains a minimum *distance* with the robot car). Both the agents choose their actions using a model predictive controller [28].

In **Robot** a simulated human corrects a collaborative robot arm. The robot's action space is its 3-DoF linear end-effector velocity. The environment includes two objects: a cup and a plate. The robot is not sure whether it should reach or avoid each object, and learns the human's preferences θ based on the human's corrections. When training the corrective term

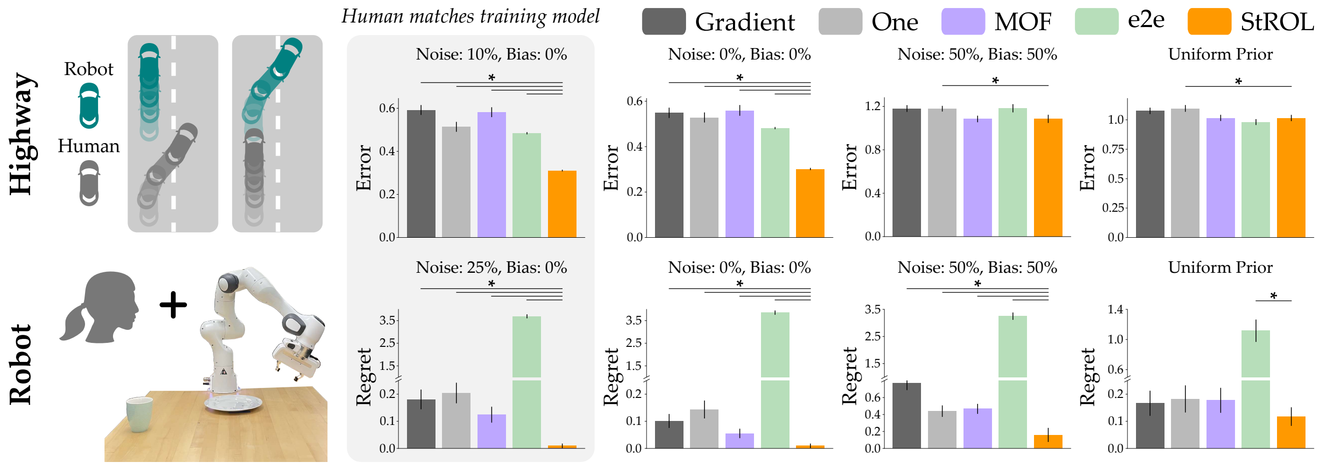

Fig. 3. Simulation results from Section V. We compare StROL to state-of-the-art baselines in a multi-agent **Highway** environment (Top) and a collaborative **Robot** setting (Bottom). In **Highway**, the robot car takes turns interacting with 250 simulated human cars and tries to predict whether it should change lanes. We measure the *Error* between the robot's learned estimate θ and the simulated human's objective θ^*. In **Robot**, 100 simulated humans teach a 7 DoF Franka-Emika robot arm to reach for or avoid two stationary objects (also see Figure 2). We measure the *Regret* over the robot's learned behavior. For both environments we simulate humans with different levels of noise and bias. During offline training, **e2e** and **StROL** expected 10% noise in **Highway** and **25%** noise in **Robot**. The left column corresponds to this training setting. The other columns compare each method as the simulated human's noise, bias, and prior over θ^* deviates from the training data. Error bars show SEM, and an $*$ represents statistical significance ($p < 0.05$).

$\hat{g}$ we assume that the human has a bimodal prior over these features: the human likely prefers to either (a) reach the plate and avoid the cup or (b) go to the cup and avoid the plate. During each interaction the simulated human corrects the robot's behavior over the first 5 timesteps. After each timestep the robot updates its preferences θ and recomputes its trajectory to optimize for the learned reward function.

Dependent Variables. We measured the accuracy of the robot's learned estimate θ in both environments. In **Highway** we recorded the *Error* between the learned reward parameters θ and the true parameters θ^*, where $Error = \|\theta^* - \theta\|$. In the competitive, multi-agent highway environment error is especially because if the robot incorrectly estimates θ, the actions taken by the robot car can lead to a collision.

In the collaborative **Robot** setting, we explore whether the robot's learned behavior aligns with the human's preferences. We measure the *Regret* across the robot's learned trajectory:

$$Regret(\xi) = \sum_{x \in \xi^*} R(x, \theta^*) - \sum_{x \in \xi_\theta} R(x, \theta^*) \tag{11}$$

Here ξ^* is the optimal trajectory for reward weights θ^* and ξ_θ is the robot's learned trajectory (i.e., the trajectory that optimizes reward parameters θ. Regret quantifies how much worse the robot's trajectory is compared to the human's ideal trajectory: lower values indicate better performance.

Simulated Humans. We simulated humans with different priors and increasing levels of suboptimality. More specifically, our simulated human chose actions according to:

$$u_h = u_{\mathcal{H}}^* + \delta, \quad \delta \sim \mathcal{N}(\epsilon, \sigma), \quad \theta^* \sim P(\theta) \tag{12}$$

where σ is controls the *Noise* and ϵ is the *Bias*. When training **StROL** and **e2e** we assumed a given level of noise and zero bias. When training in the **Highway** environment we set $\sigma = 10\%$ of the magnitude of the largest action, and in *Robot* we set $\sigma = 25\%$ of the magnitude of the largest action. Then during out experiments we performed simulations with increasing levels of noise and bias; hence, the simulated human's behavior *deviated* from the training behavior that our approach expected. Similarly, during training we set the prior as a multimodal distribution, and then during our experiments we performed tests where the human's reward parameters θ^* were sampled from a uniform prior.

Hypothesis. We had the following two hypotheses:

H1. ***StROL*** *will outperform the baselines when the human's behavior is similar to the training behavior.*

H2. *When humans act in unexpected ways,* ***StROL*** *will perform better than or comparable to the baselines.*

Results. Our results are summarized in Figure 3. First we will breakdown these results for the **Highway** environment. Across all trials and conditions, a repeated measures ANOVA found that the robot's learning algorithm had a significant effect on learning error ($F(4, 996) = 32.098$, $p < 0.05$). Looking at the error plots in Figure 3 (Row 1, Columns 2-3), when the human actions at test time are similar to the human actions during training, **StROL** significantly outperforms all the baselines ($p < 0.05$). As the noise and bias in the human's actions increase (Row 1, Column 4), each algorithm performs similarly: **StROL** is not significantly different from **Gradient** ($p = 0.051$), **MOF** ($p = 0.98$), or **e2e:** ($p = 0.80$). The same tend occurs when the simulated human's actions are sampled from an unexpected prior (Row 1, Column 5). Put together, these results suggest that — when the human driver behaves similar to our model — **StROL** leads to robust robots that accurately predict θ. In the worst case — where the human significantly deviates from the robot's expectations — **StROL** is on par with existing methods.

We found similar trends when analyzing the **Robot** results. A repeated measures ANOVA with a Greenhouse-Geisser correction ($\epsilon = 0.552$) revealed that the learning algorithm had a significant effect on the regret ($F(2.21, 218.49) = 1287.1$, $p < 0.05$). The plots in Figure 3 (Row 2, Columns 2-3) show that the robot's regret is significantly lower when the robot uses **StROL** ($p < 0.05$). As the humans become increasingly random, the regret for **StROL** increases, but it is still lower than the baselines ($p < 0.05$). On the other hand, if the human tries to teach an unexpected task that is outside the robot's prior, **StROL** performs on par with **Gradient** ($p = 0.40$), **One** ($p = 0.30$), and **MOF** ($p = 0.31$). Thus, we find support for hypotheses **H1** and **H2**.

VI. User Study

To evaluate our approach in real-world environments, we conducted an in-person user study where participants interacted with a 7-DoF Franka-Emika Panda robot arm. During each trial users attempted to teach the robot their desired reward by applying forces and torques to the robot arm. We compared StROL to state-of-the-art approaches that learn online from human interventions [3], [5]. Videos of our user study are provided here: https://youtu.be/uDGpkvJnY8g

Independent Variables. StROL leverages Algorithm 1 offline to modify the learning dynamics and expand the basins of attraction. Similar to the simulations in Sections V, our baselines include **One** [3] and **MOF** [5].

Experimental Setup. The experimental setup consisted of a 7-Dof Franka-Emika robot arm carrying a cup across a table with a plate and a pitcher of water (see Figure 1). The robot started each trial by following a randomly generated trajectory. Users then physically intervened to correct the motion of the robot arm to teach it three different tasks. For **Task 1** users taught the robot to carry the cup to the *plate*, while keeping the cup close to the *table* and away from the *pitcher*. **Task 2** was similar to **Task 1**, with the addition that the users had to teach the robot to carry the cup at the correct *orientation*. Finally, in **Task 3** the users taught the robot to move away from all objects while keeping the cup upright. Task 1 had three features ($\theta \in \mathbb{R}^3$) while Tasks 2 and 3 had four features ($\theta \in \mathbb{R}^4$). These manipulation tasks with physical human corrections were similar to the user study environments used in [5] and [3]. When training **StROL** offline the robot's multimodal prior included **Task 1** and **Task 2**, but **Task 3** involved a new region of reward parameters that the learner did not expect.

Participants and Procedure. We recruited 12 participants from the Virginia Tech community (6 female, average age 23.5 ± 3.08). Participants gave informed consent prior to the start of the experiment under Virginia Tech IRB $\#22-755$.

The participants performed all three tasks with each learning algorithm. The order of the learning algorithms was counterbalanced using a Latin square design (e.g., some participants started with **StROL**, others started with **One**, etc.). Before each task the robot played the ideal trajectory for that task (i.e., the robot showed the behavior that the participant should teach to the robot). Between each trial the robot reset from scratch: the robot did not carry over what it learned about θ from one trial to another.

We trained **StROL** offline to shape the learning dynamics. During training we used the noisy human model in Equation (12) with $\sigma = 25\%$ of action magnitude and $\epsilon = 0$. The multimodal prior $P(\theta)$ used during training consisted of 3-4 modes; these modes includes the desired behaviors for **Task 1** and **Task 2**, but not for **Task 3**. We emphasize that **StROL** was trained *offline with simulated human data*, and then deployed *online to perform zero-shot learning with real humans* and improve the overall robot performance.

Dependent Variables. We seek systems that learn the human's reward accurately and rapidly. To analyze the learning accuracy, we measured the robot's *Regret* according to Equation (11). To analyze how rapidly the robot learned, we measured the total amount of time the human spent correcting the robot arm (*Correction Time*).

We also administered a 7-point Likert scale survey to access the participants' subjective responses. Our survey questions were organized into two multi-item scales: whether the users thought the robot *learned* to perform the task correctly, and how *intuitive* it was for participants to teach the robot. Every participant completed this survey 9 times: once after they finished working with every task and algorithm.

Hypothesis. We had the following hypotheses for this study:
H3. *With* ***StROL*** *users will teach the robot more quickly (shorter correction time) and accurately (lower regret).*
H4. *Participants will find* ***StROL*** *to be a more intuitive learner as compared to the baselines.*

Results. We first explore hypothesis **H3**, and refer to the objective results portrayed in Figure 4 (Column 1-3). A Repeated Measures ANOVA revealed that robot's learning algorithm had a significant effect on the correction time ($F(2, 22) = 5.602$, $p < 0.05$) and regret ($F(1.332, 14.651) = 9.108$, $p < 0.05$). Post hoc comparisons showed that **StROL** had significantly lower correction time and regret as compared to the baselines ($p < 0.05$) (see 4 Column 1-2). Column 3 in Figure 4 shows how a scatter plot of how the regret for each learning algorithm varied with the correction time. Across all participants and tasks, we observed consistently lower regret with **StROL**. But with **One** and **MOF**, there were situations where the teacher spent a long time correcting, and the regret remained high. For these approaches, we also observed situations where the participants gave up teaching after a few corrections, leading to a short correction time and high regret. This provides support for our hypothesis **H3**.

Now to explore hypothesis **H4**, consider the Likert scale survey in Figure 4 (Column 4). After verifying that the scales used for the survey were reliable (Cronbach's $\alpha > 0.7$), we grouped the responses for each scale into a combined score. A repeated measures ANOVA ($F(2, 70) = 21.301$, $p < 0.05$) suggested that the users perceived **Our** approach to be significantly more *intuitive* than the baselines while providing corrections to the robot ($p < 0.05$). Similarly, a repeated measures ANOVA with a Huynh-Feldt correction ($\epsilon = 0.807$, $F(1.6, 56.5) = 18.1$, $p < 0.05$) revealed that after observing the robot's final behavior, the users thought **Our** approach *learned* better than the baselines ($p < 0.05$).

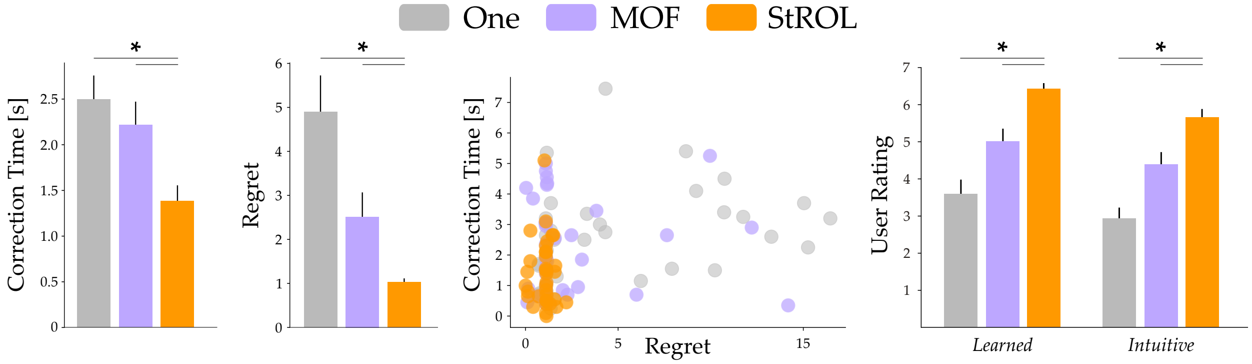

Fig. 4. Objective and subjective results from the user study in Section VI. Participants physically interacted with a 7-DoF robot arm (see Figure 1) to teach it three different tasks. The robot used StROL or other online learning methods [3], [5] to infer the human's reward parameters in real-time. (Left) We plot the time users spent correcting the robot and the regret across the robot's learned trajectory averaged over all three tasks. (Middle) For each individual task and participant (3 tasks × 12 participants) we plot their regret vs. correction time. (Right) Finally, we report the average user ratings from our 7-point Likert scale survey. Error bars show SEM and an $*$ denotes statistical significance ($p < 0.05$).

VII. Conclusion

In this paper we formalize online learning from humans as a dynamical system, and present a control theoretic perspective to enhance the convergence and robustness properties of the robot learner. Given an initial learning rule, we leverage Lyapunov stability analysis and offline training to learn a corrective term which modifies the learning dynamics and expands the basins of attraction around a multimodal prior. In simulations and a user study we show that our resulting algorithm (StROL) improves the robot's learning when interacting with suboptimal and noisy human teachers.

Limitations. Our proposed approach augmented the initial learning dynamics g with the corrective term $\hat{g}$ to reach the new learning dynamics $\tilde{g} = g + \hat{g}$. The relative weights of g and $\hat{g}$ must be tuned by the designer. If $\hat{g}$ is unbounded, the learned corrective term may constrain the robot learner into the basins of attraction, preventing the human from teaching reward parameters θ that lie outside the robot's prior. Conversely, if the designer constrains $\hat{g}$ to be too small, then StROL will not have a significant effect on the robot's learning. In our future work, we plan to leverage the noise in user's actions to tune the magnitude of $\hat{g}$ automatically.

References

[1] W. Jin, T. D. Murphey, Z. Lu, and S. Mou, "Learning from human directional corrections," *IEEE Transactions on Robotics*, 2022.

[2] D. P. Losey and M. K. O'Malley, "Learning the correct robot trajectory in real-time from physical human interactions," *ACM Transactions on Human-Robot Interaction*, vol. 9, no. 1, pp. 1–19, 2019.

[3] D. P. Losey, A. Bajcsy, M. K. O'Malley, and A. D. Dragan, "Physical interaction as communication: Learning robot objectives online from human corrections," *IJRR*, vol. 41, no. 1, pp. 20–44, 2022.

[4] A. Jain, S. Sharma, T. Joachims, and A. Saxena, "Learning preferences for manipulation tasks from online coactive feedback," *IJRR*, 2015.

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

[6] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *ICML*, 2006, pp. 729–736.

[7] M. Hagenow, E. Senft, R. Radwin, M. Gleicher, B. Mutlu, and M. Zinn, "Corrective shared autonomy for addressing task variability," *IEEE Robotics and Automation Letters*, 2021.

[8] H. J. Jeon, S. Milli, and A. Dragan, "Reward-rational (implicit) choice: A unifying formalism for reward learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4415–4426, 2020.

[9] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Expert intervention learning: An online framework for robot learning from explicit and implicit human feedback," *Autonomous Robots*, pp. 1–15, 2022.

[10] M. Tucker, E. Novoseller, C. Kann, Y. Sui, Y. Yue, J. W. Burdick, and A. D. Ames, "Preference-based learning for exoskeleton gait optimization," in *ICRA*, 2020, pp. 2351–2357.

[11] K. Kronander and A. Billard, "Online learning of varying stiffness through physical human-robot interaction," in *ICRA*, 2012.

[12] M. Khoramshahi and A. Billard, "A dynamical system approach to task-adaptation in physical human–robot interaction," *Autonomous Robots*, vol. 43, pp. 927–946, 2019.

[13] Y. Li, G. Carboni, F. Gonzalez, D. Campolo, and E. Burdet, "Differential game theory for versatile physical human–robot interaction," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 36–43, 2019.

[14] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche, "The role of roles: Physical cooperation between humans and robots," *IJRR*, vol. 31, no. 13, pp. 1656–1674, 2012.

[15] A. Broad, I. Abraham, T. Murphey, and B. Argall, "Data-driven koopman operators for model-based shared control of human–machine systems," *IJRR*, vol. 39, no. 9, pp. 1178–1195, 2020.

[16] R. Tian, M. Tomizuka, A. D. Dragan, and A. Bajcsy, "Towards modeling and influencing the dynamics of human learning," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2023.

[17] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *arXiv preprint arXiv:2102.03861*, 2021.

[18] M. A. Thornton and D. I. Tamir, "People accurately predict the transition probabilities between actions," *Science Advances*, 2021.

[19] R. Shah, D. Krasheninnikov, J. Alexander, P. Abbeel, and A. Dragan, "Preferences implicit in the state of the world," *ICLR*, 2019.

[20] R. Dubey, P. Agrawal, D. Pathak, T. L. Griffiths, and A. A. Efros, "Investigating human priors for playing video games," *arXiv preprint arXiv:1802.10217*, 2018.

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

[22] B. Zhang and H. Soh, "Large language models as zero-shot human models for human-robot interaction," *arXiv preprint arXiv:2303.03548*, 2023.

[23] A. Bajcsy, A. Siththaranjan, C. J. Tomlin, and A. D. Dragan, "Analyzing human models that adapt online," in *ICRA*, 2021.

[24] A. Rubinstein, *Modeling bounded rationality*. MIT press, 1998.

[25] T. L. Griffiths, F. Lieder, and N. D. Goodman, "Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic," *Topics in Cognitive Science*, 2015.

[26] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2008, vol. 3.

[27] Z. Cao, E. Biyik, W. Z. Wang, A. Raventos, A. Gaidon, G. Rosman, and D. Sadigh, "Reinforcement learning based control of imitative policies for near-accident driving," in *RSS*, July 2020.

[28] B. Kouvaritakis and M. Cannon, "Model predictive control," *Switzerland: Springer International Publishing*, vol. 38, 2016.