Learning from Physical Human Corrections, One Feature at a Time

Andrea Bajcsy
University of California, Berkeley
abajcsy@berkeley.edu

Dylan P. Losey
Rice University
dlosey@rice.edu

Marcia K. O'Malley
Rice University
omalleym@rice.edu

Anca D. Dragan
University of California, Berkeley
anca@berkeley.edu

ABSTRACT

We focus on learning robot objective functions from human guidance: specifically, from physical corrections provided by the person while the robot is acting. Objective functions are typically parametrized in terms of *features*, which capture aspects of the task that might be important. When the person intervenes to correct the robot's behavior, the robot should update its understanding of which features matter, how much, and in what way. Unfortunately, real users do not provide optimal corrections that isolate exactly what the robot was doing wrong. Thus, when receiving a correction, it is difficult for the robot to determine which features the person meant to correct, and which features were changed unintentionally. In this paper, we propose to improve the efficiency of robot learning during physical interactions by reducing *unintended* learning. Our approach allows the human-robot team to focus on learning *one feature at a time*, unlike state-of-the-art techniques that update all features at once. We derive an online method for identifying the single feature which the human is trying to change during physical interaction, and experimentally compare this one-at-a-time approach to the all-at-once baseline in a user study. Our results suggest that users teaching one-at-a-time perform better, especially in tasks that require changing multiple features.

KEYWORDS

physical human-robot interaction, learning from demonstration, human teachers

ACM Reference Format:
Andrea Bajcsy, Dylan P. Losey, Marcia K. O'Malley, and Anca D. Dragan. 2018. Learning from Physical Human Corrections, One Feature at a Time. In *Proceedings of 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI '18).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3171221.3171267

1 INTRODUCTION

Consider a household situation in which a robot and a human work in close physical proximity. While performing its task, the robot does something wrong, and the human intervenes to physically correct the robot as it is moving. For example, the robot is moving a fragile cup from a cabinet to the table, and a nearby human notices that the robot is carrying the cup too high above the table: if the cup were to drop from that height, it would likely break!

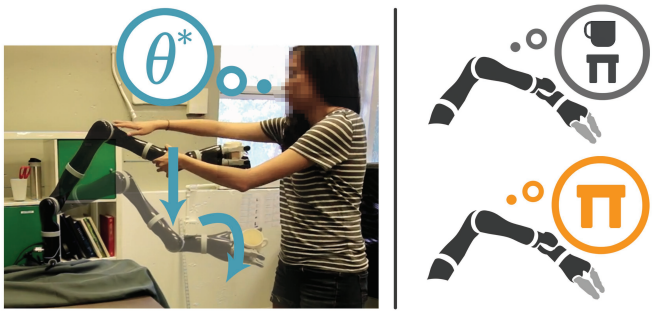


Figure 1: Participant pushes on the robot to teach it to go closer to the table. In the process of giving this correction, the human changes both the robot's distance from table and — inadvertently — the orientation of a cup which the robot is grasping (blue arrows). Typically, the robot would learn about both cup and table features from this one correction (top right). We propose that robots interacting with humans should learn about only *one feature at a time* (bottom right).

To correct the robot's behavior, the human intuitively pushes the robot's end-effector towards the table to signal their motion preference. Ideally, the human's correction will only affect the cup's distance from the table; in practice, however, human actions are noisy and imperfect [7, 19, 20, 24], especially when kinesthetically maneuvering robotic manipulators while trying to carefully orchestrate their multiple degrees of freedom [1]. As a consequence, when the person pushes down on the end-effector, they accidentally change not only the robot's distance from the table, but also the orientation of the cup (see Fig. 1).

This single human interaction has therefore adjusted two task features: the cup's distance from the table and the cup's orientation. From the robot's perspective, it is not immediately clear what the person actually intends: do they (a) want the robot to carry the cup closer to the table, or do they (b) additionally want the robot to carry the cup at a new orientation?

State-of-the-art algorithms default to the latter interpretation. Prior work has built on Inverse Reinforcement Learning (IRL) [13, 16–18, 24] to formalize learning from physical human corrections as an estimation problem: the robot estimates the objective function that it should optimize during the task by treating human corrections as evidence about the objective function's parameters[1].

[1]Similar to prior IRL work, we will assume that the correct features for the task have been identified *a priori*, and are known to both the human and the robot.

HRI '18, March 5–8, 2018, Chicago, IL, USA

ACM ISBN 978-1-4503-4953-6/18/03…$15.00
https://doi.org/10.1145/3171221.3171267

Under the ideal objective function parameters, the corrected behavior has to have a lower cost than the robot's current behavior [3, 11]. Therefore, when the person's correction changes multiple features — however slightly — a rich hypothesis space will lead to the robot updating its understanding about the importance of *all* of these features (top right in Fig. 1).

This traditional approach works well with perfect or near-perfect corrections; however, with real people come aspects of corrections that are not always intended. These unintended corrections lead to *unintended learning*. In other words, the robot attempts to learn from and alter its behavior based on all the inputs, even those that are superfluous. Returning to our previous example, if all features were updated the robot would learn (correctly) that the cup should be lower, and (incorrectly) that the cup should be carried at a different orientation. In general, because of the inherent physical difficulty in simultaneously correcting many degrees of freedom of a robotic arm, learning about all features at once may systematically cause the robot to infer more from the human's corrections than desired.

Our insight is that we can alleviate unintended robot learning by focusing the learning on only one feature at a time.

For tasks where the human is attempting to change the importance of just one feature, this insight helps the robot reject inadvertent adjustments on the other features (bottom right in Fig. 1). But even for tasks in which the human wants to correct several features, learning one feature at a time enables people to break down the task and teach *sequentially*. Indeed, sequential teaching may come more naturally to people collaborating with robots [21, 22], and reduces the burden on users to coordinate all aspects of the task simultaneously during each individual correction.

Based on our insight, we make the following contributions:

Online Feature Identification. As the robot is executing its task, the human collaborator can intervene and provide physical corrections. We formulate the problem of identifying which one feature the person is trying to correct at each time step, derive a solution, and justify a simple approximation for online performance. We hypothesize that this approach will result in a better learning process, with a more accurate objective function being inferred by the robot at each time step, and a better final outcome.

User Study Testing One-at-a-Time Learning. After validating our algorithm in 2-D simulations with an approximately optimal human, we put our hypothesis to the test in a user study on a 7-DoF robotic manipulator. These experiments compare one-at-a-time and all-at-once learning within a factorial design, across tasks that need just one feature to be corrected, and tasks that need multiple features to be corrected. We find that one-at-a-time learning is especially helpful in the second case, where the person's teaching task is more complex. People also prefer it, finding that the robot is better at understanding their corrections and requires less reteaching.

Overall, our work provides a practical improvement for learning objective functions online from physical human-robot interaction.

2 ONE-AT-A-TIME OBJECTIVE LEARNING FROM PHYSICAL HUMAN INTERACTION

2.1 Why Learn from Physical Corrections?

When a human and robot are collaborating in close proximity, physical interaction — in which the human touches, pushes, pulls, or otherwise guides the robot — is almost inevitable. The way in which a robot responds to such physical human-robot interaction (pHRI) depends on *how* the robot interprets those corrections.

Traditionally, the human's interactions are treated in one of three ways [9]: as disturbances to be rejected [5, 12, 23], as collisions to be detected and avoided [4], or as operator signals to be followed by switching into a compliant mode [8, 10, 14]. In all cases, the robot does not *learn* from the human's actions; once the human stops interacting, the robot resumes its original behavior.

In contrast, we argue that interactions are *intentional*, and therefore *informative* — the human interacts with the robot because it is doing something wrong, and the human's correction indicates how the robot should behave. Furthermore, since the way in which the robot chose its behavior was by optimizing an objective function, interaction suggests that this objective function was incorrect. Thus, rather than stubbornly continuing to optimize the same wrong objective, the robot should instead leverage the human's feedback in order to update its understanding of the objective function.

2.2 Learning Problem Statement

Assume the robot starts in some configuration q^0 at time $t = 0$. Let Ξ be the space of trajectories beginning at q^0 and ending at a feasible goal configuration, where each $\xi \in \Xi$ is a sequence of configurations. Next, let $\Phi : \Xi \rightarrow \mathbb{R}^F$ be a vector-valued function mapping trajectories to feature values, with $\Phi_i(\xi)$ signifying the value of the i-th feature.

Similar to prior IRL work [13, 16, 18, 24], the robot's objective function (here a cost function) is parametrized by $\theta \in \mathbb{R}^F$, which weights the importance of these features along the entire trajectory:

$$C(\xi) = \theta \cdot \Phi(\xi) \tag{1}$$

The robot starts off with an initial objective function θ^0 at time $t = 0$, and optimizes this objective function to produce its initial trajectory:

$$\xi^0 = \arg\min_{\Xi} \theta^0 \cdot \Phi(\xi) \tag{2}$$

After identifying ξ^0, the robot starts to execute this initial trajectory.

The person interacting with the robot has some desired objective function that they want the robot to optimize, denoted as θ^*. The robot does not have access to these parameters — they are internal to the person (and here assumed to be constant). However, at every time step t, the person might intervene to move the robot away from its current configuration by some Δq^t. The robot should then treat the human's correction Δq^t as an observation about θ^*, and update its objective from θ^t to θ^{t+1}, such that this new objective function is closer to θ^*.

2.3 All-at-Once Learning

Following [3], we interpret the change in configuration Δq^t as an indication of the corrected trajectory, ξ_c^t, that the human would prefer for the robot to execute:

$$\xi_c^t = \xi^t + M^{-1}(0, .., \Delta q^t, ...0)^T \tag{3}$$

Here ξ^t is the robot's current trajectory — optimal under θ^t — and M is a matrix that smoothly propagates the local correction Δq^t along the rest of the trajectory [6].

Next, based on [11] and [18], we make the core assumption that the corrected trajectory ξ_c^t is better than the current trajectory ξ^t with respect to the ground truth θ^*. Recalling that our objective

function is a cost function, this implies:

$$\theta^* \cdot \Phi(\xi_c^t) < \theta^* \cdot \Phi(\xi^t) \tag{4}$$

To now find a θ^{t+1} closer to θ^*, we select a weight vector that is both (a) near the current θ^t and (b) maximally makes (4) hold:

$$\theta^{t+1} = \arg\min_{\theta \in \Theta} \theta \cdot (\Phi(\xi_c^t) - \Phi(\xi^t)) + \frac{1}{2\alpha}||\theta - \theta^t||^2 \tag{5}$$

Note that $\alpha > 0$. This optimization problem is a quadratic in θ, so we will take the gradient of (5) and set it equal to 0:

$$\nabla_\theta = \Phi(\xi_c^t) - \Phi(\xi^t) + \frac{1}{\alpha}(\theta - \theta^t) = 0 \tag{6}$$

Rearranging (6), we finally obtain:

$$\theta^{t+1} = \theta^t - \alpha(\Phi(\xi_c^t) - \Phi(\xi^t)) \tag{7}$$

Interestingly, (7) is the same update rule from co-active learning [11] and online maximum margin planning [18], shown by [3] to be an approximate solution to the partially observable Markov decision process that treats θ^* as the hidden state and optimizes the cost parametrized by θ^*. This update rule has an intuitive interpretation: if a feature has a higher value in corrected trajectory than in the current trajectory, (7) decreases corresponding weight — making it lower-cost — and thus encourages the optimizer to generate subsequent trajectories where that feature also has a higher value.

Under this method, the robot updates the weights on *all* features that the person changed with their correction during the current time step.

2.4 One-at-a-Time Learning

A natural solution for restricting the number of learned features might be to switch the regularization term in (5) to the L_1 norm [13, 15], which encourages sparsity of the weight update. However, there is no guarantee that this will result in changing just one weight; it may still update all the features that the human corrected, including those that were accidentally changed.

In this work, to capture one-at-a-time learning, we now make a different assumption about $\bar{\xi}_c^t$, the *intended* corrected trajectory. While the actual corrected trajectory, ξ_c^t, might change multiple features, we assume that the human's *intended* corrected trajectory, $\bar{\xi}_c^t$, changes only a single feature.

We simplify the intended corrected trajectory into an intended change in features, $\Delta\Phi_c^t$, and impose the constraint that $\Delta\Phi_c^t$ can only have one non-zero entry: this entry represents the feature which the person wants to update. Note that our one-at-a-time strategy does *not* mean that only one feature ever changes throughout the task. Instead, at every time step t there can be a *different* intended feature change, and so the person can sequentially change the weights to match their desired objective over multiple corrections.

Without loss of generality, assume that the human is attempting to change the i^{th} entry in θ^t, the robot's current feature weights. If the human interacts to only update the weight on the i^{th} feature, then their correction of the robot's current trajectory, ξ^t, should change the feature count in the direction $J(\theta_i) = \frac{\partial\Phi(\xi^t)}{\partial\theta_i^t}$. In other words, given that the person is an optimal corrector and that their interaction was meant to change just the weight on the i^{th} feature, then we would expect them to correct the trajectory such that they produce a feature difference exactly in the direction $J(\theta_i)$.

Realistically, however, human corrections are noisy — even for expert users [2] — and will not necessarily induce the optimal feature difference during every correction. Despite these imperfections, we assume that the result of their correction will still noisily optimize the distance (dot product) in the optimal direction. This provides us with an *observation model*, from which we can find the likelihood of observing a specific feature difference given the one feature which the human is attempting to update:

$$P(\Delta\Phi|i) \propto e^{J(\theta_i)\cdot\Delta\Phi} \tag{8}$$

Accordingly, for the observed feature difference $\Delta\Phi = \Phi(\xi_c^t) - \Phi(\xi^t)$, the feature which the human is most likely trying to change is:

$$\begin{aligned} i^* &= \arg\max_i P(\Phi(\xi_c^t) - \Phi(\xi^t)\big|i) \\ &= \arg\max_i J(\theta_i) \cdot (\Phi(\xi_c^t) - \Phi(\xi^t)) \end{aligned} \tag{9}$$

Using (9), we can estimate *which* feature the person wanted to update during their physical correction. Next, by leveraging i^* and the observed feature difference, we can reconstruct $\Delta\Phi_c^t$, the human's *intended* feature difference. Recall that — if the human wanted to only update feature i^* — their intended feature difference would ideally be in the direction $J(\theta_{i^*}) = \frac{\partial\Phi(\xi^t)}{\partial\theta_{i^*}^t}$, and so we can choose $\Delta\Phi_c^t \propto J(\theta_{i^*})$. In practice, however, we will simplify this derivative by projecting the actual feature difference induced by the human's interaction onto the i^{*th} axis, $\Delta\Phi_c^t = (0, .., \Phi_{i^*}(\xi_c^t) - \Phi_{i^*}(\xi^t), ..0)^T$. Thus, once we have identified which feature the person most wants to change during their current interaction, i^*, we argue that the intended feature correction should only change this one feature [2].

Evaluating $J(\theta_i)$ requires numerical differentiation, i.e., finding an optimal trajectory at least $F + 1$ times at each time step (where F is the number of features). To make this process run in real-time, we approximate $J(\theta_i)$ as proportional to $(0, .., 1, ..0)^T$. In other words, we assume that when the i^{th} weight changes, it predominantly causes a change in the i^{th} feature along the corresponding optimal trajectory. Substituting this simplification back into (2.4), we have reduced our method for finding the feature which the human intends to change into a simple, yet intuitive, heuristic: only the feature that changed the *most* as a result of the human's correction should be updated. We note, however, that this heuristic has its roots in the more principled approach that was detailed above. Our update rule now becomes

$$\theta^{t+1} = \theta^t - \alpha\Delta\Phi_c^t \tag{10}$$

Overall, isolating a single feature at every time step is meant to prevent unintended learning. If the person is trying to correct multiple features, they can still do so: the robot will pick up on what seems like the most dominant feature in the correction, adjust that, and then give the person a chance to correct whatever remains during the next time step. Due to the noisy nature of human corrections, we hypothesize that this one-at-a-time update strategy will lead to shorter trajectories through the learned weight space — which reach the ideal weight more directly — when compared to a strategy that tries to update everything at once. In what follows,

[2] To ensure that all features are equally sensitive, we normalized each feature by the maximal attainable feature difference by computing optimal trajectories offline with a range of θ values.

we first show some simulation analysis with optimal and noisy humans, and then test our hypothesis in a user study.

3 SIMULATIONS

In order to better validate and compare the all-at-once and one-at-a-time learning methods described in Section 2, we conducted human-robot interaction simulations. These simulations show that updating one feature per interaction can help prevent unintended learning, particularly when the human interacts sub-optimally.

Setting. We will consider a vertical planar environment, where the y-axis corresponds to height above a table and the x-axis is parallel to that table. The simulated robot is attempting to move from a fixed start position, s, to a fixed goal position, g. The robot is modeled as a single point, and the robot's configuration is its current (x, y) position. A simulated human is standing beside the table near the start position, and physically interacts with the robot to correct its behavior when necessary.

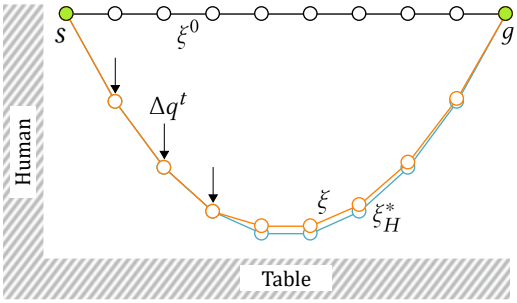
The robot does not know the true feature weights of the human's objective function, θ^*, but the robot does know that there are three different features which the human might care about: the length of the robot's trajectory (*length*), the robot's height above the table (*table*), and the robot's distance from the human (*human*). Here the *table* feature corresponds to the height along the y-axis, since the table is a surface at $y = 0$, and the *human* feature corresponds to the distance along the x-axis, since the human is standing at $x = 0$. The weight of the *length* feature is fixed, and the robot learns the relative weights associated with *table* and *human* features over the course of the task. The human's true reward parameter is $\theta^* = [0.5, 0]$, where 0.5 is the true weight associated with *table* and 0 is the true weight associated with *human*. Initially, the robot believes that $\theta^0 = [0, 0]$, and so the robot is unaware that it should move closer to the table.

Simulated Human. We consider two different simulated humans: (a) an *optimal human*, who corrects the robot to exactly follow their desired trajectory and (b) a *noisy human*, who imperfectly corrects the robot's trajectory.

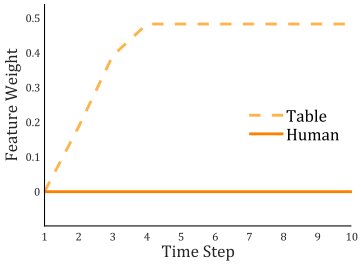
At the start of the task, the optimal human identifies a desired trajectory: $\xi_H^* = \arg\min_{\Xi} \theta^* \cdot \Phi(\xi)$. During the task, the human does not change ξ_H^*, and interacts with the robot to make it follow this desired trajectory. At each time step t the human provides a correction Δq^t that changes the robot's current configuration to the desired configuration, $\xi_H^*(t)$, but the human only provides this correction if the robot's distance from $\xi_H^*(t)$ is greater than some acceptable margin of error.

In contrast, the noisy human takes actions sampled from a Gaussian distribution: these actions are centered at the optimal human action with a bias in the x-direction. This bias introduces a systematic error, where the noisy human accidentally pulls the robot closer to their body when attempting to significantly correct the vertical *table* feature. As a result of this noise and bias, the noisy human may unintentionally correct the *human* feature.

Analysis. We performed two different simulations: one with the optimal human (see Fig. 2), and one with the noisy human (see Fig. 3). When the human optimally corrects the robot's *table* feature in Fig. 2, they never unintentionally affect the weight of the *human* feature, and so all-at-once and one-at-a-time learning both yield the exact same results for the optimal human.

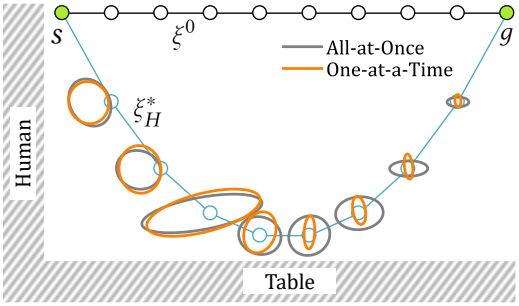


(a) Optimal human: Teaching the robot

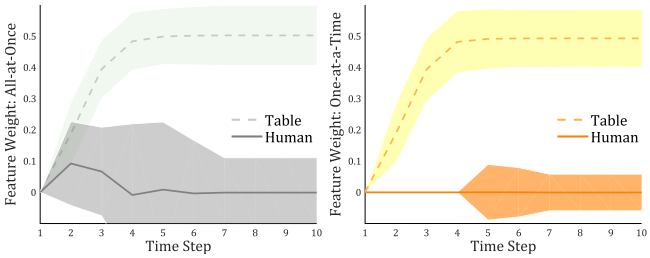


(b) Optimal human: Learning weights

Figure 2: Simulation with optimal human. (a) Human corrects the robot during the first few time steps, and the robot follows the human's desired trajectory afterwards. (b) The robot's estimated feature weights converge to the human's true feature weights.

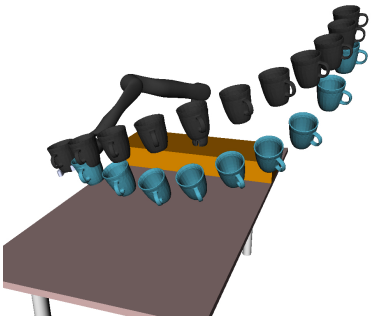


(a) Noisy human: Teaching the robot

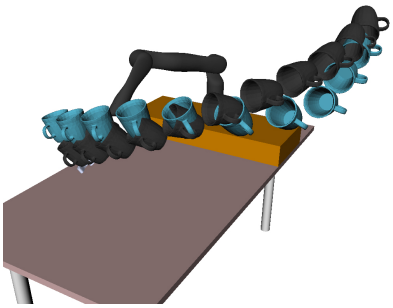


(b) Noisy human: Learning feature weights

Figure 3: Simulation with noisy human. (a) The human noisily corrects the robot's trajectory, where the ellipses show the robot's states with 95% confidence over 100 simulations. (b) With all-at-once, the robot initially learns that the *human* feature is important, and the person must undo that unintended learning. One-at-at-time learning reduces the unintended effects of the human's noisy corrections; this causes the robot to converge towards the human's desired trajectory more rapidly.



(a) Task 1: Correct one feature, the distance to table



(b) Task 2: Correct two features, the cup orientation and distance to table

Figure 4: Depictions of the robot trajectories for each of the two experimental tasks. The black path represents the original trajectory and the blue path represents the human's desired trajectory.

By contrast, the noisy human unintentionally corrects the *human* features at the start of the task (when trying to correct the *table* features), and, as such, we observed different behavior for all-at-once and one-at-a-time learning in Fig. 3. Although the robot follows a similar mean trajectory for both learning methods, and eventually converges to the correct feature weights in each case, we observe that all-at-once had a longer learning process and more persistent human interaction. In particular, the length of the mean path in feature space from θ^0 to θ^T was 0.57 for all-at-once vs. 0.49 for one-at-a-time; the length of the mean path specifically for the *human* feature weight was 0.23 for all-at-once vs. 0.001 for one-at-a-time. Recall that the robot was constrained to reach its goal position in 10 steps; we found that, in the all-at-once case, the human interacted with the robot during an average of 5.24 steps, and, in the one-at-a-time case, the human interacted with the robot during an average of 3.56 steps.

These simulations showcase that, when the human interacts sub-optimally, their corrections can lead to unintended learning on the robot's part, which the human must then exert additional effort to undo. For the simulation we have described, updating only one feature per time step helps to mitigate accidental learning, demonstrating the potential benefits of our proposed one-at-a-time learning method.

4 EXPERIMENTS

We conducted an IRB-approved user study to investigate the benefits of one-at-a-time learning. During each experimental task, the robot began with a number of incorrect weights in its objective, and the participants intervened to physically correct the robot.

4.1 Independent Variables

We use a 2 by 2 factorial design. We manipulated the *learning strategy* with two levels, all-at-once and one-at-a-time, as well as the *number of feature weights that need correction*, one feature weight and all the feature weights.

In the all-at-once learning strategy, the robot updated all the feature weights from a given interaction with the gradient update from Equation (7) and then replanned a new trajectory with the updated weights. In the one-at-a-time condition, the robot chose the feature that changed the most using Equation (2.4), updated according to Equation (10), and then replanned a new trajectory withe the updated θ.

4.2 Dependent Measures

4.2.1 Objective. To analyze the objective performance of the two learning strategies, we split the objective measures into four categories:

Final Learned Reward: These measure how closely the learned reward matched the optimal reward by the end of the trajectory.

We measured the dot product between the optimal and final reward vector, denoted $DotFinal = \theta^* \cdot \theta^T$. We also analyzed the regret of the final learned reward, which is the weighted feature difference between the ideal trajectory and the learned trajectory

$$RegretFinal = \theta^* \cdot \Phi(\xi_{\theta^*}) - \theta^* \cdot \Phi(\xi_{\theta^T})$$

and the individual feature differences between the ideal reward and the trajectory induced by the final learned reward

$$TableDiffFinal = |\Phi_{Tb}(\xi_{\theta^*}) - \Phi_{Tb}(\xi_{\theta^T})|$$
$$CupDiffFinal = |\Phi_C(\xi_{\theta^*}) - \Phi_C(\xi_{\theta^T})|$$

Learning Process: Measures about the learning process, i.e. $\hat{\theta} = \{\theta^0, \theta^1, \ldots, \theta^T\}$, included the average dot product between the true reward and the estimated reward over time: $DotAvg = \frac{1}{T}\sum_{i=0}^{T} \theta^* \cdot \theta^i$. We also measured the length of the $\hat{\theta}$ path through weight space for both cup, $\hat{\theta}_C$, and table, $\hat{\theta}_{Tb}$ weights. Finally, we computed the number of times the cup and table weights were updated away from the optimal θ^* (denoted by *CupAway* and *TableAway*).

Executed Trajectory: For the actual executed trajectory, ξ_{act}, we measured the regret

$$Regret = \theta^* \cdot \Phi(\xi_{\theta^*}) - \theta^* \cdot \Phi(\xi_{act})$$

and the individual table and cup feature differences between the ideal and actual trajectory

$$TableDiff = |\Phi_{Tb}(\xi_{\theta^*}) - \Phi_{Tb}(\xi_{act})|$$
$$CupDiff = |\Phi_C(\xi_{\theta^*}) - \Phi_C(\xi_{act})|$$

Interaction: Interaction measures on the forces applied by the human, $\{u_H^0, u_H^1, \ldots, u_H^T\}$, included the total interaction force, $IactForce = \sum_{t=0}^{T} ||u_H^t||_1$ and total interaction time.

4.2.2 Subjective. For each condition, we administered a 7-point Likert scale survey about the participant's interaction experience (see Table 1 for questions). We separated our survey questions into four scales: success in teaching the robot about the task, correctness of update, needing to undo corrections because the robot learned something wrong, and ease of undoing.

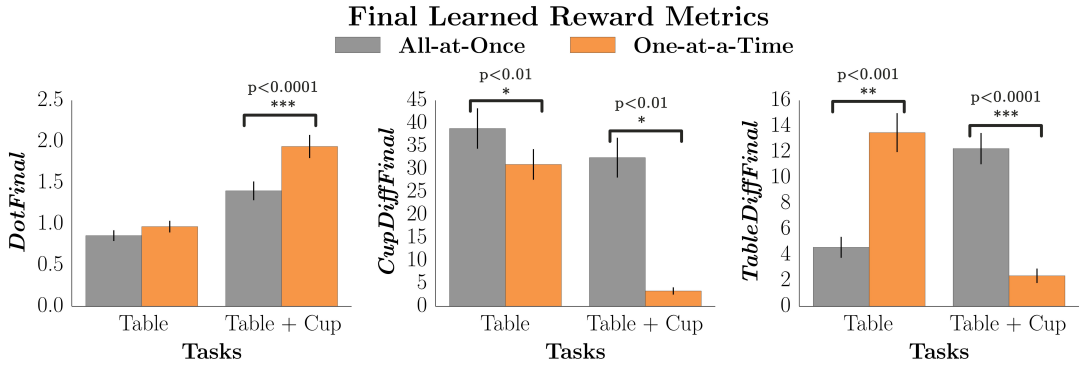


Figure 5: The final learned weight vector with one-at-a-time is closer to the ideal weight vector for the task where two feature weights are incorrect (left). Looking at the individual feature differences from ideal: while the final cup weight is closer to ideal for one-at-a-time for both tasks (center), the ideal table weight is actually significantly *further away* from the ideal for the one-at-a-time strategy during the one-feature task (right). However, for the two feature task, the one-at-a-time method outperforms the all-at-once for final learned cup and table weights.

4.3 Hypotheses

H1. *Updating one feature at a time significantly increases the final learned reward, enables a better learning process, results in lower regret for the executed trajectory, and leads to less interaction effort and time compared to all-at-once update.*

H2. *Participants will perceive the robot as more successful at accomplishing the task, correctly updating its knowledge of the task, less likely to learn about extraneous aspects of the task, and be easier to correct if it did learn something wrong in the one-at-a-time condition.*

4.4 Tasks

We designed two experimental household manipulation tasks for the robot to perform in a shared workspace (see Fig.4 for setup). For each experimental task, the robot carried a cup from a start to end pose with *an initially incorrect objective*. One of the tasks focused on participants having to correct a *single aspect of the incorrect objective*, while the other needed them to correct *all parts* of the objective. Participants were instructed to physically intervene to correct the robot's behavior during the task. Similar to state-of-the-art methods, all the features in the robot's objective were chosen to be intuitive to a human to ensure that participants could understand how to correct the robot.

In Task 1, the robot's objective had only *one feature weight incorrect*. The robot's default trajectory took a cup from the participant and put it down on the table, but carried the cup too far above the table (top of Fig.4). In Task 2, *all the feature weights started out incorrect* in the robot's objective. The robot also took a cup from the participant and put it down on the table, but this time it initially grasped the cup at the wrong angle and was also carrying the cup too high above the table (bottom of Fig.4).

4.5 Participants

We used a within-subjects design and counterbalanced the order of the conditions during experiments. In total, we recruited 12 participants (7 female, 4 male, 1 non-binary trans-masculine, aged 18-30) from the campus community, 11 of which had technical backgrounds and 1 of which did not. None of the participants had experience interacting with the robot used in our experiments.

4.6 Procedure

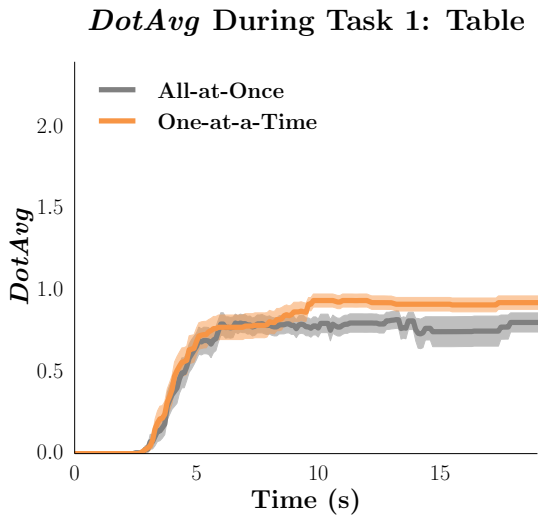
Before beginning the experiment, participants performed a familiarization task to become comfortable teaching the robot with physical corrections. The robot's original trajectory moved a cup from a shelf to a table, but the robot did not initially care about tilting the cup mid-task. The robot's objective contained only one aspect of the task (cup orientation) and participants had to correct only this one aspect. Afterwards, for each experimental task, the participants were shown the robot's default trajectory as well as what their desired trajectory looks like. They were also told what aspects of the task the robot is always aware of (cup orientation and distance of end-effector to table) as well as which learning strategy they were interacting with. Participants were told the difference between the two learning strategies in order to minimize in-task learning effects. Note, however, that we did *not* tell participants to teach the robot in any specific style (like one aspect as a time), only about how the robot reasons about their corrections.

4.7 Analysis

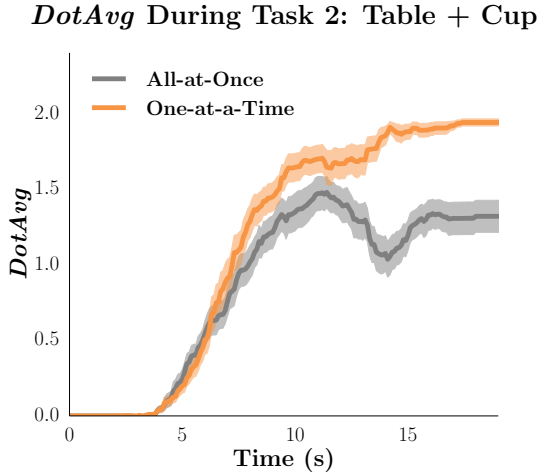
4.7.1 Objective. **Final Learned Reward.** We ran a factorial repeated-measures ANOVA with learning strategy and number of features as factors, and user ID as a random effect, for each of the measures capturing the quality of the final learning outcome. Fig.5 summarizes our findings about the final learned weights for each learning strategy.

For the final dot product with the true reward, we found a significant main effect of the learning strategy ($F(1, 81) = 29.86$, $p < .0001$), but also an interaction effect with the number of features ($F(1, 81) = 13.07$, $p < .01$). The post-hoc analysis with Tukey HSD revealed that one-at-a-time led to a higher dot product on the two feature task ($p < .0001$), but there was no significant difference on the one-feature task (where one-at-a-time led to slightly higher dot product).

We next looked at the final regret, i.e. the difference between the cost of the learned trajectory and that of the ideal trajectory. For this metric we found an interaction effect, suggesting that one-at-a-time led to lower regret for the two-feature task but not for the one-feature task. Looking separately at the feature values for table and cup, we found that one-at-a-time led to a significantly lower difference for the cup feature across the board ($F(1, 81) = 11.30$,



(a) In the task with only one wrong feature weight, there is no significant difference between the two methods in average dot product over time.



(b) In contrast to (a), when two feature weights are wrong, the one-at-a-time strategy outperforms the all-at-once strategy when it came to a higher dot product over the duration of the trajectory.

Figure 6: The one-at-a-time strategy shows significantly more consistent alignment between the estimated weight vector, θ^t, and the ideal weight vector, θ^*, than the all-at-once for the two feature task. This indicates that when multiple aspects of the objective need changing, the one-at-a-time method enables more accurate learning.

$p < .01$, no interaction effect), but that one-at-a-time only improved the difference for the table on the two feature task ($p < .0001$) – it actually significantly increased the difference on the one feature task ($p < .001$).

Overall, we see that one-at-a-time learns something significantly better across the board for the two-feature task. When it comes to the one feature task, the results are mixed: it led to a significantly better result for the cup orientation, but significantly worse for the table distance feature.

Learning Process. For the average dot product between the estimated and true reward over time, our analysis revealed almost identical outcomes to before, when we were looking at the final reward only (see Fig.6).

We also found that one-at-a-time resulted in significantly fewer updates in the wrong direction for the cup weight across the board ($F(1, 81) = 44.91$, $p < .0001$) and for the table weight ($F(1, 81) = 22.02$, $p < .0001$), with no interaction effect. Fig.7 highlights these findings and their connection to the subjective metrics.

Looking at the length of the path through the space of weights, we found a main effect of learning strategy ($F(1, 81) = 26.82$, $p < .0001$), but also an interaction effect ($F(1, 81) = 6.55$, $p = .01$). The posthoc analysis with Tukey HSD revealed that for the the one-feature task, one-at-a-time resulted in significantly shorter path traversed through weight space ($p < .0001$). The path was shorter with the two-feature task as well, but the difference was not significant. The effect was mainly due to the one-at-a-time method resulting in a shorter path for the cup weight on the one-feature task, as revealed by the posthoc analysis ($p < .0001$).

Overall, we see that the quality of the learning process was significantly higher for the one-at-a-time strategy across both tasks. When one aspect and all aspects of the objective were wrong, one-at-a-time led to fewer wrong weight updates and resulted in the learned reward across time being closer to the true reward.

The Executed Trajectory. We found no significant main effect of the learning strategy on the regret of the executed trajectory: the two strategies lead to relatively similar actual trajectories with respect to regret. Both regret as well as the feature differences from ideal for cup and table showed significant interaction effects.

Interaction Metrics. We found no significant effects on interaction time or force.

Summary of Objective Metric Analysis. Taken together, these results indicate that a one-at-a-time learning strategy leads to a better learning process across the board. On the more complex two-feature task, this strategy also leads to unquestionably better learning outcomes. For the one-feature task, learning one feature at a time enables users to better avoid the wrong perturbation of the correct weight (on the cup feature), but is not as good as the all-at-once method at enabling users to properly correct the wrong weight (on the table feature). Thus, H1 was partially supported: although updating one feature weight at a time does not improve task performance when there is only one aspect of the objective wrong, reasoning about one feature weight at a time leads to significantly better learning and task performance when all aspects of the objective are wrong.

4.7.2 Subjective. We ran a repeated measures ANOVA on the results of our participant survey. After testing the reliability of our 4 scales, we found that the correct update and undoing scale were significantly reliable, so we grouped these into a combined score (see Chronbach's α in Table 1). We analyzed success and undoing ease separately as they were not reliable.

For the correct update scale, we found a significant effect of learning strategy ($F(1, 33) = 5.09, p = 0.031$), showing that participants perceived the one-at-a-time strategy as better at updating the robot's objective according to their corrections. Additionally, the undoing scale showed a significant effect of learning strategy ($F(1, 33) = 10.35, p < 0.01$), with the one-at-at-time strategy being less likely to learn the wrong thing and cause the participants to have to undo a correction. For ease of undoing, when analyzing Q9 and Q10 individually we found no significant effect of strategy.

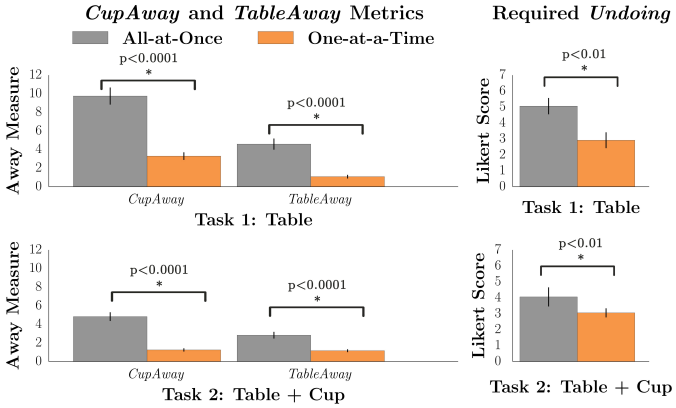


Figure 7: The one-at-a-time strategy results in significantly less weight updates that are away from the optimum weight across all tasks (left top, left bottom). These findings are consistent with the subjective likert data from the undoing scale, where participants perceived the one-at-a-time method as less likely to learn the wrong thing and need an additional undoing action.

Summary of Subjective Metric Analysis. The subjective data echoes some of the objective data results. Participants perceived that one-at-a-time better understood their corrections and required less undoing due to unintended learning, partially supporting H2.

5 DISCUSSION

In this paper, we compared the performance of one-at-a-time and all-at-once learning for two tasks: one that required correcting a single feature, and another that required correcting multiple features of a robot's objective. For the multiple feature task, learning about one feature at a time was objectively superior: it led to a better final learning outcome (Fig.5), took a shorter path to the optimum, and had fewer incorrect inferences and undoings along the way (Fig.6). However, the results were not as clear for the single feature task: the one-at-a-time method lessened unintended learning on the weights that were initially correct, but it hindered learning for the incorrect weights. However, participants subjectively preferred the one-at-a-time strategy overall: they thought it was better at learning the correct aspects of the task and required less undoing.

We hypothesize that the superior objective performance of the one-at-a-time strategy in the second task is due to the increased complexity of the teaching task. It appears that one-at-a-time learning is more useful as the teaching task becomes more complex and requires fixing more aspects of the robot's objective. However, with simple teaching tasks that only require one aspect of the objective to change, it is not yet clear whether one-at-a-time is a significantly better learning strategy.

5.1 Limitations and Future Work

It is both a limitation and a strength that we chose the simplest possible feature selection method for the one-at-a-time task. On the one hand, this is an intuitive and computationally inexpensive method to examine as a first exploration into teaching robot objectives online via physical interaction. At the same time, our simple learning strategy was not consistently superior in the simple task. This opens the door for analyzing more sophisticated methods that perform Bayesian inference on the intended feature, or low-pass filtering to prevent high frequency changes in which features gets

Table 1: Likert scale questions were grouped into four categories: success in accomplishing the task, correctness of update (reliable), needing to undo corrections because of unintended learning (reliable), and ease of undoing.

	Likert Questions	Cronbach's α
succ	Q1: I successfully taught the robot how to do the task.	–
correct update	Q2: The robot correctly updated its understanding about aspects of the task that I did want to change. Q3: The robot wrongly updated its understanding about aspects of the task I did NOT want to change. Q4: The robot understood which aspects of the task I wanted to change, and how to change them. Q5: The robot misinterpreted my corrections.	.84
undoing	Q6: I had to try to undo corrections that I gave to the robot, because it learned the wrong thing. Q7: Sometimes my corrections were just meant to fix the effect of previous corrections I gave. Q8: I had to re-teach the robot about an aspect of the task that it started off knowing well.	.93
undo ease	Q9: When the robot learned something wrong, it was difficult for me to undo that. Q10: It was easy to re-correct the robot whenever it misunderstood a previous correction of mine.	.66

updated to improve overall learning and usability. Additionally, while our method worked well with intuitive features like "distance to table", additional work is needed to investigate how well each method works when the features are non-intuitive to the human.

Perhaps our largest limitation in this work is our demographics: our study participants were primarily individuals with a technical background (with one exception). Future work must consider a more diverse user population to ensure external validity.

Not only do we need algorithms that can learn from humans, but the methods must also reason about the difficulties humans experience when trying to kinesthetically teach a complex robotic system. To simplify the teaching process, we propose that robots should learn one aspect of the objective at a time from physical corrections. While our user studies indicate the benefits of this method, it is only a first step towards seamless human-robot interaction.

REFERENCES

[1] 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.
[2] 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.
[3] Andrea Bajcsy, Dylan P Losey, Marcia K O'Malley, and Anca D Dragan. 2017. Learning robot objectives from physical human interaction. In *Conference on Robot Learning (CoRL)*.
[4] Alessandro De Luca, Alin Albu-Schaffer, Sami Haddadin, and Gerd Hirzinger. 2006. Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on*. IEEE, 1623–1630.
[5] 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.
[6] Anca D Dragan, Katharina Muelling, J Andrew Bagnell, and Siddhartha S Srinivasa. 2015. Movement primitives via optimization. In *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 2339–2346.
[7] Thomas L Griffiths, Falk Lieder, and Noah D Goodman. 2015. Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in Cognitive Science* 7, 2 (2015), 217–229.
[8] 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 *Intelligent Robots and Systems, IEEE/RSJ International Conference on*. IEEE, 3356–3363.
[9] Sami Haddadin and Elizabeth Croft. 2016. Physical human–robot interaction. In *Springer Handbook of Robotics*. Springer, 1835–1874.
[10] Neville Hogan. 1985. Impedance control: An approach to manipulation; Part II—Implementation. *Journal of Dynamic Systems, Measurement, and Control* 107, 1 (1985), 8–16.
[11] 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.
[12] Nathanaël Jarrassé, Themistoklis Charalambous, and Etienne Burdet. 2012. A framework to describe, analyze and generate interactive motor behaviors. *PloS one* 7, 11 (2012), e49945.
[13] Mrinal Kalakrishnan, Peter Pastor, Ludovic Righetti, and Stefan Schaal. 2013. Learning objective functions for manipulation. In *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 1331–1336.
[14] 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.
[15] Andrew Y Ng. 2004. Feature selection, $L1$ vs. $L2$ regularization, and rotational invariance. In *Machine Learning, International Conference on*. ACM, 78.
[16] Andrew Y Ng, Stuart J Russell, et al. 2000. Algorithms for inverse reinforcement learning. In *Machine Learning, International Conference on*. 663–670.
[17] Deepak Ramachandran and Eyal Amir. 2007. Bayesian inverse reinforcement learning. *Urbana* 51, 61801 (2007), 1–4.
[18] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. 2006. Maximum margin planning. In *Machine Learning, International Conference on*. ACM, 729–736.
[19] Ariel Rubinstein. 1998. *Modeling bounded rationality*. MIT Press.
[20] Jonathan Sorg, Satinder P Singh, and Richard L Lewis. 2010. Internal rewards mitigate agent boundedness. In *Machine Learning, International Conference on*. 1007–1014.
[21] Andrea L Thomaz and Cynthia Breazeal. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence* 172, 6-7 (2008), 716–737.
[22] Andrea L Thomaz and Maya Cakmak. 2009. Learning about objects with human teachers. In *Human Robot Interaction, ACM/IEEE International Conference on*. ACM, 15–22.
[23] Chenguang Yang, Gowrishankar Ganesh, Sami Haddadin, Sven Parusel, Alin Albu-Schaeffer, and Etienne Burdet. 2011. Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE Transactions on Robotics* 27, 5 (2011), 918–930.
[24] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning. In *AAAI*.