MIND MELD: Personalized Meta-Learning for Robot-Centric Imitation Learning

Mariah L. Schrum*, Erin Hedlund-Botti*, Nina Moorman, Matthew C. Gombolay
Georgia Institute of Technology
Atlanta, United States
{mschrum3, ehedlund6, ninamoorman}@gatech.edu, matthew.gombolay@cc.gatech.edu

***Abstract*—Learning from demonstration (LfD) techniques seek to enable users without computer programming experience to teach robots novel tasks. There are generally two types of LfD: human- and robot-centric. While human-centric learning is intuitive, human centric learning suffers from performance degradation due to covariate shift. Robot-centric approaches, such as Dataset Aggregation (DAgger), address covariate shift but can struggle to learn from suboptimal human teachers. To create a more human-aware version of robot-centric LfD, we present Mutual Information-driven Meta-learning from Demonstration (MIND MELD). MIND MELD meta-learns a mapping from suboptimal and heterogeneous human feedback to optimal labels, thereby improving the learning signal for robot-centric LfD. The key to our approach is learning an informative personalized embedding using mutual information maximization via variational inference. The embedding then informs a mapping from human provided labels to optimal labels. We evaluate our framework in a human-subjects experiment, demonstrating that our approach improves corrective labels provided by human demonstrators. Our framework outperforms baselines in terms of ability to reach the goal ($p < .001$), average distance from the goal ($p = .006$), and various subjective ratings ($p = .008$).**

***Index Terms*—Learning from demonstration, personalization, meta-learning**

I. Introduction

Learning from Demonstration (LfD) seeks to enable humans to teach robots new skills via human task demonstrations without the need for users to have prior experience in computer programming [2]. In LfD, the robot learns a policy that maps the state of the world to how the robot should act to accomplish the human-specified or demonstrated task [36]. Researchers have pursued two principle types of LfD: human-centric and robot-centric [22]. In human-centric LfD, a human typically performs the task, and the robot infers from this demonstration the task specification. An example of human-centric LfD is Behavioral Cloning (BC), i.e. *mimicry* [11], where the robot records the human demonstration of the task and uses supervised learning to learn a policy mapping states to actions. However, BC suffers from covariate shift issues due to a mismatch between the distribution of states given by the demonstration versus those experienced by the robot when attempting to accomplish the task [26], [31], [32].

*Authors contributed equally.

This work was supported by Georgia Institute of Technology State Funding, a NASA Early Career Fellowship under grant 80HQTR19NOA01-19ECF-B1, MIT Lincoln Laboratory, a gift from Konica Minolta, and the National Science Foundation under grants 1545287 and 20-604.

Robot-centric LfD is an alternative to human-centric LfD and addresses the problem of covariate shift [32] by instead learning from a human's corrective feedback signal at each time step as the robot executes the task [22]. One example of robot-centric LfD is Dataset Aggregation (DAgger) [32]. Ross et al. showed that learning from human corrective actions solves the problem posed by covariate shift [32]. Many robot-centric, as well as human-centric, LfD algorithms assume the demonstrator is an expert at the task and that they will provide optimal demonstrations or feedback [29]. When the demonstrator is a Wizard-of-Oz oracle [30] and provides optimal demonstrations, prior work has shown that DAgger can learn policies that are more sample efficient and accurate than human-centric LfD algorithms [32]. However, these studies may not translate to real-world settings where non-oracle, heterogeneous human demonstrators provide sub-optimal demonstrations [1], [4], [22], [42]. Prior work has shown that humans struggle to provide high quality corrective actions during robot-centric LfD [39]. Additionally, humans are heterogeneous: the way humans provide feedback may differ depending upon the individual's abilities and prior experience [28], [35]. Therefore, robot-centric LfD approaches need to account for the teacher's suboptimality and heterogeneity to learn effective policies. However, prior work fails to take into account demonstrator suboptimality and human heterogeneity in robot-centric LfD.

To fill this gap, we aim to harness the potential advantages of robot-centric algorithms (i.e., increased policy performance and sample efficiency) and improve upon robot-centric algorithms by explicitly learning to account for heterogeneity and suboptimality in teaching. We introduce Mutual Information-driven Meta-learning from Demonstration (MIND MELD), which uses a Long Short-Term Memory (LSTM) neural network-based architecture to meta-learn a person-specific mapping from human-provided, corrective-action labels to idealized labels, which are inferred based upon a distribution of calibration tasks with known, optimal labels. Because human feedback is heterogeneous, we propose to use variational inference to learn a personalized embedding that encapsulates information about a person's style of providing corrective feedback. We then use the personalized embedding to map each individual's suboptimal labels to labels that more closely approximate optimal labels, thereby improving the performance of robot-centric LfD algorithms. Optimal labels (i.e., ground truths) are only necessary for a small set of calibration tasks [14], [17] to learn to improve upon human labels and are not needed at test time.

In this paper, we conduct an IRB-approved within-subjects study, comparing the performance of MIND MELD to a robot-centric baseline, DAgger, and a human-centric baseline, BC. We evaluate these algorithms based on their ability to learn the task of driving an autonomous vehicle to a goal without collisions as well as various subjective metrics. Additionally, we analyze how the learned personalized embeddings capture the demonstrator's style and improve suboptimal labels.

In our work, we contribute the following:

1) We formulate MIND MELD, a novel, personalized LfD framework for improving upon suboptimal corrective labels by inferring individual demonstrator styles.
2) We demonstrate that MIND MELD objectively outperforms prior work in a human-subjects experiment in its ability to reach the goal more often than BC ($p < .001$) and DAgger ($p < .001$).
3) We show that users prefer MIND MELD over DAgger and BC in terms of trust ($p < .001$), workload ($p = .005$), perceived intelligence ($p = .008$), and likeability ($p = .004$).

II. Related Works

Prior work has explored human-centric LfD for learning a robot policy for task execution from an expert human demonstrator [2], [12], [24], [29], [32]. The simplest and most ubiquitous form of human-centric learning is BC, in which a robot infers the mapping from states to actions via supervised learning based on human demonstrations [18], [31]. However, if the learner deviates from the demonstrated path, covariate shift occurs due to a mismatch between the states induced by the demonstrations and those experienced by the robot when rolling out a policy. Due to this covariate shift, a learner's mistake count can compound quadratically with regards to the time horizon [32].

In response to this problem, Ross et al. introduced Dataset Aggregation (DAgger), a robot-centric LfD approach that aggregates a training data set of expert labels queried during policy rollout [32]. DAgger utilizes the state distribution induced by the current policy to solicit labels from the expert and employs a gating function to determine the mixture of expert and learner during each rollout. Ross et al. proved linear-loss, no-regret guarantees and showed that with high-quality, expert demonstrations, DAgger outperforms prior work.

However, Laskey et al. [22] showed that robot-centric learning approaches, such as DAgger, can lead to human mislabelling, resulting in poor learner performance. Additionally, DAgger requires a heavy workload from the demonstrator, which can result in demonstrator fatigue and poor training results [21], [23], [27]. Prior work has attempted to reduce the amount of corrective feedback required of the demonstrator by DAgger to improve teacher-learner interaction [16], [21], [25], [42]. He et al. proposed an imitation-learning-by-coaching algorithm in which the learner must imitate actions of progressively increasing difficulty [16]. In this approach, task loss is reduced by demonstrating to the learner preferable actions. Results have shown that this coaching scheme can outperform DAgger and achieve a lower regret bound when the demonstrator is an oracle, but no study has been conducted demonstrating this method's advantage with human teachers.

In related work, Kelly et al. proposed to reduce expert workload while improving upon expert-provided demonstrations through Human Gated DAgger (HG-DAgger), allowing the expert to decide when to provide feedback via a gating function [21]. HG-DAgger learns a stationary policy such that labels are obtained via a policy that stabilizes around expert trajectories. Spencer et al. expanded on this idea, utilizing both information about when the expert does and does not intervene, in the Expert Intervention Learning (EIL) algorithm [42]. HG-DAgger and EIL both focus on augmenting *when* the human should provide feedback during a trajectory, whereas our approach focuses on *how*, by improving the feedback itself. Because our approach is complimentary and orthogonal to robot-centric LfD approaches such as HG-DAgger and EIL, these approaches are not suitable benchmarks. Instead, our approach could be used in conjunction with these and other related approaches to improve upon human-provided labels.

Knox and Stone developed TAMER, which allows humans to provide feedback in the form of a scalar reward [5]. TAMER accounts for delayed feedback, but does not account for heterogeneous demonstrators. Other approaches, such as T-Rex and D-Rex, use inverse reinforcement learning (IRL) to improve upon poor human demonstrations by learning a reward function from a set of ranked demonstrations [6], [7]. Also using IRL, Chen et al. introduced SSRR to learn from suboptimal demonstrations by characterizing the relationship between noise and performance [9]. However, there is a lack of prior work accounting for both the heterogeneity and suboptimality of humans for robot-centric LfD. Therefore, there is a need for LfD algorithms that can effectively learn from the typical, non-expert human demonstrator in a robot-centric paradigm [29]. Our approach is the first to improve upon robot-centric learning by inferring demonstrator style via personalized embeddings to correct for suboptimal demonstrations. We maintain the advantages of robot-centric learning (i.e., reducing covariate shift) while making robot-centric LfD more human-aware by accounting for the suboptimality and heterogeneity of human demonstrators.

III. Methodology

In the following section, we provide an overview of the preliminaries of our work and describe our MIND MELD algorithm for improving robot-centric LfD with suboptimal human demonstrators. We discuss our network architecture, personalized embeddings, and the mapping of suboptimal labels to more effective labels.

A. Preliminaries

The LfD problem can readily be framed as a Markov Decision Process sans reward function (MDP$\backslash$R). The MDP$\backslash$R

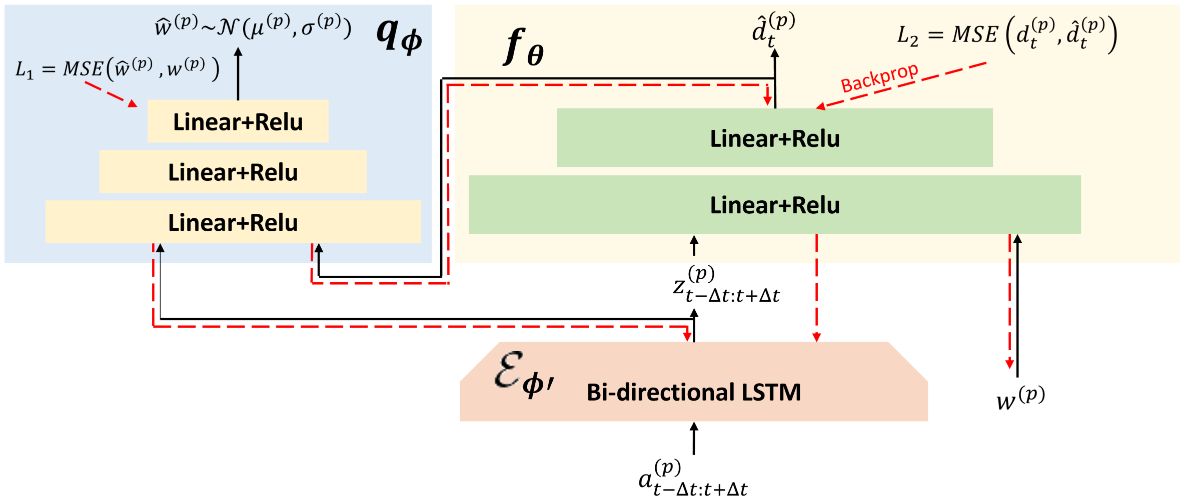


Fig. 1: This figure shows the MIND MELD network architecture. $a_t^{(p)}$ represents demonstrator p's corrective label, at time t. The recreation subnetwork, q_ϕ, maximizes mutual information between the learned embedding, $w^{(p)}$, the encoding, $z_{(t-\Delta t:t+\Delta t)}^{(p)}$, and the output, $\hat{d}_t^{(p)}$. The objective is to minimize the mean squared error (MSE) between the predicted difference, $\hat{d}_t^{(p)}$, and the true difference, $d_t^{(p)} = a_t^{(p)} - o_t$, of the demonstrator's corrective feedback and the ground truth label, o_t. We pass in the sequence of corrective feedback, $a_{(t-\Delta t:t+\Delta t)}^{(p)}$, from time $t - \Delta t$ to $t + \Delta t$ to the bi-directional LSTM and extract sequential information to inform the predictions of ground truth label at time t.

is defined by the 4-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma \rangle$. $\mathcal{S}$ represents the set of states and $\mathcal{A}$ the set of actions. $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S}' \rightarrow [0, 1]$ is the transition function that returns the probability of transitioning to state, s', from state, s, applying action, a. γ weights the discounting of future rewards. Reinforcement learning seeks to synthesize a policy, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, mapping states to actions to maximize the future expected reward. In an LfD paradigm, a demonstrator provides a set of trajectories, $\{(s_t, a_t), \forall t \in \{1, 2, ...T\}\}$, from which the agent learns a policy.

We make the following assumptions in our work.

- In the context of robot-centric learning from demonstration, humans provide corrective feedback that is suboptimal (e.g., with respect to an optimal, minimum-jerk, collision-free trajectory planner).
- These human-specified, heterogeneous, sub-optimal strategies can be represented by a learned embedding.
- Across different tasks, humans provide predictable and consistent, albeit suboptimal, corrective feedback.
- We have access to a distribution of calibration tasks from which we can obtain the optimal, ground truth labels.

Given these assumptions, we learn an individual's corrective "style" via a personalized embedding trained over a set of calibration tasks to represent the human's suboptimal tendencies. We then utilize this embedding to condition a meta-learned mapping from suboptimal corrective labels to ground truth labels given a set of calibration tasks. Our approach is a type of meta-learning as we learn an architecture over a distribution of tasks and participants in order to more effectively learn a specific LfD task.

B. Architecture

Depicted in Fig. 1 is the architecture of our network, which consists of three components: 1) the bidirectional LSTM encoder, $\mathcal{E}_{\phi'} : A \rightarrow Z$, 2) the prediction subnetwork, $f_\theta : Z \times W \rightarrow \mathbb{R}$, and 3) the mutual information subnetwork, $q_\phi : Z \times \mathbb{R} \rightarrow \mathcal{N}_W$. The label we aim to improve upon is $a_t^{(p)}$. We denote the set of d-dimensional, personalized embeddings as W, and the set of k-dimensional encodings extracted from the sequences of corrective feedback as $Z \subset \mathbb{R}^k$. $\mathcal{E}_{\phi'}$ is trained to extract the encoding, $z_{(t-\Delta t:t+\Delta t)}^{(p)} \in Z$, for the sequence of corrective labels, $a_{(t-\Delta t:t+\Delta t)}^{(p)}$, provided by person p from time $t - \Delta t$ to $t + \Delta t$.

f_θ maps the encoding, $z_{(t-\Delta t:t+\Delta t)}^{(p)}$, and personalized embedding, $w^{(p)}$, to the difference, $d_t^{(p)} = o_t - a_t^{(p)}$, between the ground truth label (obtained via a controller such as MPC [8] or Stanley [41]) and the individual's corrective label, where $d_t^{(p)} \in \mathbb{R}^k$. The subnetwork q_ϕ learns a mapping of the encoding, $z_{(t-\Delta t:t+\Delta t)}^{(p)}$, and predicted difference, $\hat{d}_t^{(p)}$, to a posterior distribution over the demonstrator's embedding, $w^{(p)}$. We initialize $w^{(p)}$ based upon the prior, $\hat{w}^{(p)} \sim \mathcal{N}(0, 1)$, and obtain an estimate of the individual's learned embedding, $\hat{w}^{(p)}$, by sampling from the approximate posterior.

C. Variational Inference

This work is motivated by the assumption that humans are not optimal or homogeneous in how they provide feedback, thus necessitating democratized LfD methods which account for both heterogeneity and suboptimality. Note that we handle the fact that individuals' demonstrations are suboptimal and heterogeneous separately. We capture information about an individual's corrective "style" (i.e., *how* they are suboptimal) using a personalized embedding, $w^{(p)}$, for individual p, which we then use to correct the individual's suboptimal and heterogeneous demonstrations, as described in Eq. 1. In our work, we seek to maximize the mutual information between the corrective mapping, $\hat{d}_t^{(p)}$, our learned personalized embedding, $w^{(p)}$, and the encoding of the demonstrator labels, $z^{(p)}_{(t-\Delta t:t+\Delta t)}$, such that the uncertainty of our learned embedding decreases, given informative corrective feedback.

Maximizing mutual information necessitates access to an intractable posterior distribution, $P[w^{(p)}|z^{(p)}_{(t-\Delta t:t+\Delta t)}, \hat{d}_t^{(p)}]$. Thus, we train $w^{(p)}$ to capture salient information about an individual's style by utilizing the variational lower bound, $L_I(f_\theta, q_\phi)$, as derived in Chen et al. [10] and shown in Eq. 1, where the mutual information between $z^{(p)}_{(t-\Delta t:t+\Delta t)}$, $\hat{d}_t^{(p)}$ and personalized embedding, $w^{(p)}$, is $I(w^{(p)}; z^{(p)}_{(t-\Delta t:t+\Delta t)}, \hat{d}_t^{(p)})$.

$$\begin{aligned} I(w^{(p)}; z^{(p)}_{(t-\Delta t:t+\Delta t)}, \hat{d}_t^{(p)}) &= H(w^{(p)}) - H(w^{(p)}|z^{(p)}_{(t-\Delta t:t+\Delta t)}, \hat{d}_t^{(p)}) \\ &\geq \mathbb{E}[log(q_\phi(w^{(p)}|z_{(t-\Delta t:t+\Delta t)}, \hat{d}_t^{(p)}))] + H(w^{(p)}) = L_I(f_{\theta, q_\phi}) \end{aligned} \quad (1)$$

Our network is trained by combining two loss functions: one to learn the embedding, $w^{(p)}$, and one to learn the difference, $\hat{d}_t^{(p)}$, as shown in Fig. 1. L_1 minimizes the MSE between the sampled embedding approximation, $\hat{w}^{(p)}$, and the personalized embedding, $w^{(p)}$ (equivalent to maximizing the log-likelihood of the posterior). L_2 minimizes the MSE between the predicted difference, $\hat{d}_t^{(p)}$, and the true difference, $d_t^{(p)} = o_t - a_t^{(p)}$. We backpropagate the sum of these losses (Eq. 2) to learn the embedding during training such that the personalized embedding reflects the individual's feedback style. Then, at test time, we freeze the network parameters, θ, ϕ, and ϕ' and utilize this personalized embedding to inform the mapping of demonstrator feedback.

$$L_{(\theta,\phi,\phi',w)} = L_{1_{(\theta,\phi,\phi')}} + \lambda L_{2_{(\theta,\phi')}} \quad (2)$$

$$L_{1_{(\theta,\phi,\phi')}} = \frac{1}{K+1}\sum_{k=0}^{K} ||\hat{w}_k^{(p)} - w_k^{(p)}|| \quad (3)$$

$$L_{2_{(\theta,\phi')}} = ||d_k^{(p)} - \hat{d}_k^{(p)}|| \quad (4)$$

IV. Synthetic Experiment and Pilot Study

We conduct a synthetic study [37] to demonstrate MIND MELD's ability to correct for suboptimal, heterogeneous feedback. In our synthetic experiment, we create artificial Wizard-of-Oz rollouts, ground truths, and human demonstrators. We demonstrate that the embeddings learn a meaningful representation of demonstrator stylistic tendencies (Fig. 2).

We additionally conducted an IRB approved pilot study [37] to test MIND MELD's ability to learn meaningful embeddings and improve upon suboptimal corrective feedback. After recruiting 34 participants, we found that MIND MELD was able to improve corrective feedback and learn embeddings that significantly correlate with demonstrators' stylistic tendencies, i.e., the way in which they deviate from optimal ($p < .001$).

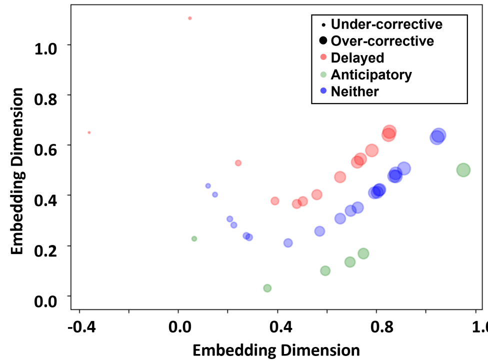


Fig. 2: This figure shows learned embeddings from our synthetic experiment. Diameter represents individuals' tendency to over-/under-correct, while color represents the tendency to provide anticipatory or delayed feedback.

Algorithm 1 MIND MELD Procedure

For M training participants, collect calibration task data
Perform gradient descent on $\theta, \phi, \phi', \omega$ until convergence (Eq. 2)
Freeze architecture parameters, ϕ, ϕ' and θ
for p in test participants **do**
  Initialize $w^{(p)} \leftarrow \frac{1}{M}\sum_{i=0}^{M} w^{(i)}$
  Collect calibration task data from p
  Perform gradient descent on ω until convergence (Eq. 4)
  Obtain initial demonstration from p.
  Present LfD algorithm conditions {MIND MELD, BC, and DAgger} in randomized order.
  for c in conditions **do**
    Train learner via condition, c, for N demonstrations.
  end for
end for

Based on results of our pilot study, we redesigned our study to better capture the stylistic tendencies of demonstrators and expanded upon our participant pool.

V. Human-Subjects Experiment

We evaluate our architecture via a human-subjects experiment with human demonstrators. Through this experiment, we demonstrate MIND MELD's ability to outperform prior LfD work by improving upon a user's suboptimal corrective feedback. Our human-subjects experiment consists of a training phase and a testing phase as discussed below. The steps comprising our study are illustrated in Algorithm 1. Our study has been approved by Georgia Tech's IRB.

Calibration Phase - In the calibration phase, we recruit participants to complete a set of calibration tasks to meta-learn the MIND MELD parameters, θ, ϕ, and ϕ' and personalized embeddings, $w^{(p)}$. Additionally, participants in this phase complete the pre-study questionnaires to capture prior experience and other demographic information.

Testing Phase - For the testing phase, we recruit a set of testing participants for a within-subjects study. These participants first complete the calibration tasks to learn their personalized embedding via Eq. 2. The participants then train

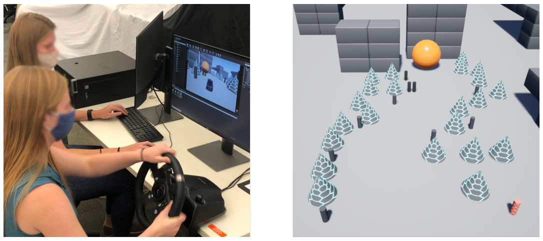


Fig. 3: The simulator and steering wheel in our human-subjects experiment are on the left and the test task is on the right.

an LfD agent via the three learning algorithms, MIND MELD, BC, and DAgger, the order of which is randomized and counterbalanced to mitigate confounding factors (e.g., fatigue, learning effects, etc.). The test task differs from the calibration tasks but is similar and falls within the same distribution (depictions of the calibrations tasks are in the Appendix). The participants in the testing phase complete both the pre-study and post-study questionnaires.

A. Driving Simulator Domain

We evaluate our approach with a human-subjects experiment in a virtual driving environment, a common domain in prior LfD, HRI, and robotics research [23], [32]. We choose to use the AirSim [40] driving simulator, an Unreal Engine-based high-fidelity physics simulator. Individuals in this experiment interact with the virtual driving environment using an Xbox steering wheel, shown in Fig. 3. We use a geometric Unreal environment where the LfD objective is to teach the agent to drive to a large, orange ball while avoiding all obstacles. The learning algorithms do not have access to the location of obstacles or the orange ball. We constrain the action space to be the position of the wheel, ranging from -540 degrees to 540 degrees. We define the state space to be composed of an image captured by a camera positioned at the front of the car as well as the car's acceleration, velocity, and position.

B. Calibration Tasks and Ground Truths

We create a series of sixteen Wizard-of-Oz [30] rollouts which are representative of successful and unsuccessful trajectories and allow us to capture the feedback styles of participants. All participants complete these tasks so MIND MELD can infer their personalized embeddings, w.

To determine ground truth optimal states for each point along the trajectories of the calibration tasks, we employ RRT* [20] (see Appendix for an example). We then apply an MPC controller along the path to determine the ground truth label at each time step.

C. Conditions

The participants first complete a set of calibration tasks which are used to learn their personalized embeddings for MIND MELD. Then, participants provide an initial demonstration from which all three agents learn an initial policy, π_0. All agents are trained for N demonstrations. Each participant experiences the following conditions in a random order.

Supervised Behavioral Cloning (BC) - Participants in this condition teach the agent via BC. To mirror our other conditions, the agent's policy is rolled out with each iteration of training so that the participant can observe the agent's behavior before providing the next demonstration.

DAgger - Participants in this condition teach the agent via vanilla DAgger [32] implemented based on prior work [22], [33]. The agent rolls out policy, π_n, and participants provide corrective feedback. The corrective labels are aggregated with the initial demonstration and corrective feedback from trials 1 to $n-1$ and the agent is retrained to yield policy, π_{n+1}.

MIND MELD (Ours) - For each demonstration, n, participants provide corrective feedback to the agent. This corrective feedback is mapped to predicted ground truth labels via MIND MELD. The mapped labels are aggregated with the initial demonstration and mapped labels from trials 1 to $n-1$ and the agent is retrained to yield policy, π_{n+1}.

D. Metrics

Below we discuss the metrics by which we evaluate MIND MELD and the learned embeddings. Both training and testing participants complete the pre-study questionnaires to determine if demographic information correlates with the learned embeddings. Only testing participants complete the post-study questionnaires. The surveys detailed below comply with the design guidelines outlined in Schrum et al. [38] and are validated from prior work when possible. The full text of the surveys and additional surveys that are not relevant to our results can be found in the Appendix. We report Cronbach's alpha (α) for each scale.

Objective Metrics

Stylistic tendencies - We analyzed participants' suboptimality by calculating their stylistic tendencies via dynamic time warping (DTW) [34] between the participant labels, a, and ground truths, o, along two-dimensions: 1) over-/under-correcting (i.e., turning the wheel too far or not enough) and 2) providing delayed/anticipatory feedback. Additional details on our calculations can be found in [37].

Goal Consistency - We measure the total number of times the agent reaches the goal, the number of demonstrations required for the agent to reach the goal, and the probability of each agent reaching the goal after each demonstration.

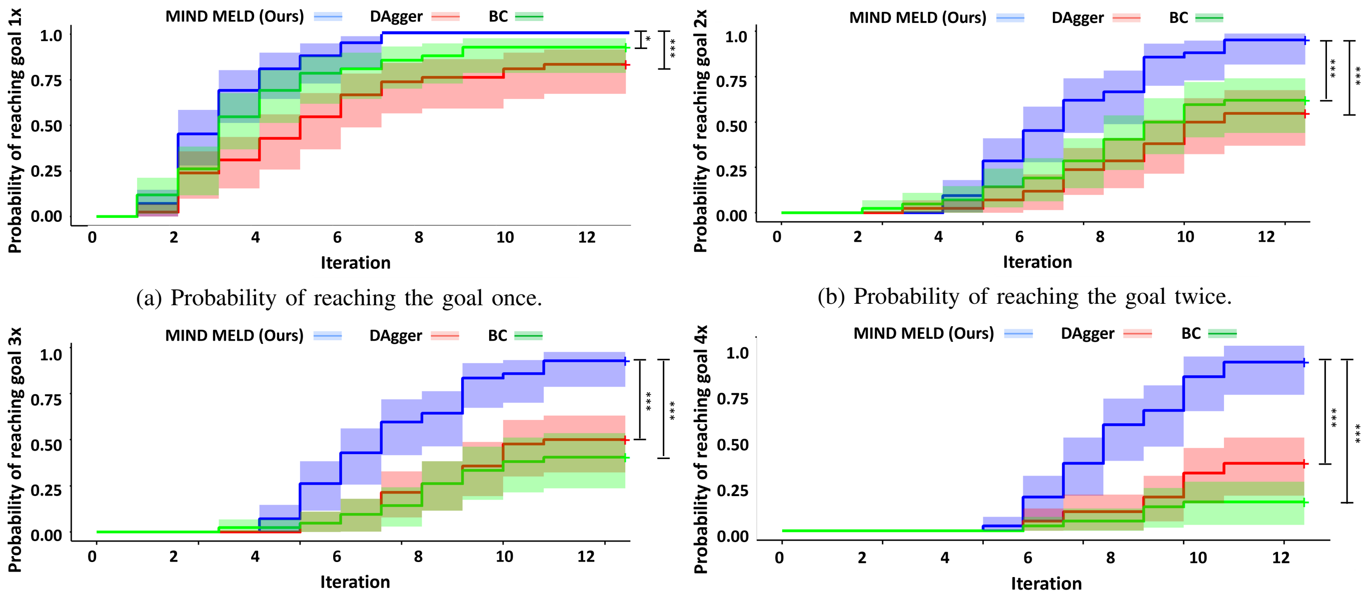
Distance - For each policy rollout of the agent, we measure the final distance between the agent and the goal.

Pre-Study Questionnaires

Prior Experience - We collect information about a participant's familiarity and experience playing video games (Cronbach's $\alpha = .93$) and driving a physical car ($\alpha = .93$) via two Likert scales to determine if prior experience correlates with the learned embeddings. Each Likert scale has eight items and a 5-point response format (strongly disagree to strongly agree). Since this survey on prior experience is ad hoc, the Appendix includes a factor analysis to validate the scales.

Post-Study Questionnaires

Trust ($\alpha = .96$) - We measure the participant's trust of the agent after each trial and for each condition [19]. In our results,



(a) Probability of reaching the goal once.

(b) Probability of reaching the goal twice.

(c) Probability of reaching the goal three times.

(d) Probability of reaching the goal four times.

Fig. 4: This figure shows that MIND MELD has a statistically significantly higher probability of reaching the goal once (Fig 4a), twice (Fig 4b), three times (Fig 4c), and four times (Fig 4d) throughout the duration of the study compared to the baselines.

we analyze the final trust survey from each condition due to the statistical testing considerations detailed in the Appendix.

Workload - We measure the workload after each condition via the NASA Task Load Index (TLX) [15].

Likeability ($\alpha = .95$) - We measure likeability after each condition via the Godspeed likeability subscale [3].

Intelligence ($\alpha = .95$) - We also measure the perceived intelligence of the agent after each condition via the intelligence subscale of Godspeed [3].

E. Procedure

An overview of our procedure for learning the MIND MELD architecture and validating MIND MELD's ability to outperform our baselines is detailed in Alg. 1. We first recruit 76 training participants by word of mouth and mailing lists. The training participants provide corrective feedback for each pre-recorded rollout which we then use to train MIND MELD and learn the parameters of MIND MELD's three subnetworks, θ, ϕ, and ϕ' as well as learn the personalized embedding, $w^{(p)}$, via Eq. 2-4. All training participants additionally answer the pre-study questionnaires.

We then recruited 42 different testing participants who experience each of the conditions discussed in Section V-C. To learn their personalized embeddings, all participants complete the calibration tasks. We then present each of the conditions discussed in Section V-C in a randomized order. All testing participants complete the pre- and post-study questionnaires.

To ensure that participants are familiar with the system before providing corrective feedback, all participants drive around in the simulator for several minutes. Additionally, participants practice providing corrective feedback in the first four calibration tasks which are not used in the training of MIND MELD so as to reduce novelty effects.

F. Hypotheses

Hypothesis 1 - *MIND MELD will improve the corrective labels provided by the participants in the calibration tasks.* We hypothesize that MIND MELD will learn to map suboptimal labels to labels that more closely approximate optimal labels by learning an embedding of stylistic tendencies of individuals.

Hypothesis 2 - *The learned embeddings will correlate with participants' stylistic tendencies and prior experience.* Based on our pilot study [37] illustrating that the learned embeddings correlated with stylistic tendencies, we predict that we will be able to reproduce these results with a larger participant pool. We also predict that the embeddings will correlate with participants' experience with video games and driving.

Hypothesis 3 - *MIND MELD will outperform DAgger and BC in terms of ability to reach goal.* We hypothesize that, due to MIND MELD's ability to correct for suboptimal feedback, MIND MELD will be more likely to reach the goal and achieve a shorter average distance from the goal.

Hypothesis 4 - *The amount by which a participant deviates from the optimal feedback style will correlate with MIND MELD's ability to outperform DAgger.* We hypothesize that participants who provide feedback that differs most from optimal (i.e., greatly over-correct) will produce poor results for DAgger. Because MIND MELD can correct for this suboptimality, the advantage of our MIND MELD algorithm over DAgger will increase with increasingly suboptimal feedback.

Hypothesis 5 - *We hypothesize that MIND MELD will achieve higher ratings on our subjective metrics compared to baselines.* Because MIND MELD corrects for suboptimality, we hypothesize that MIND MELD will be rated higher in terms of perceived intelligence, likeability, workload, and trust.

VI. Results

We recruited 76 training participants ($M = 22.8$; $SD = 5.5$; 31.2% Female), each of whom completed the calibration tasks and filled out the pre-study questionnaires. We then recruited 42 testing participants ($M = 22.1$; $SD = 2.72$; 40% Female), each of whom completed the calibration tasks, all questionnaires, and experienced the three conditions. In our following analysis, we first determine if the data complies with parametric test assumptions before employing a parametric test. Additionally, we test each model for ordering effects and confounding factors from our covariates and find none. Specific details for all parametric testing assumptions and covariates can be found in the Appendix.

We first test if our findings support **Hypothesis 1** which predicts that MIND MELD will improve upon the corrective labels provided in the calibration tasks. We find a 55% improvement in the labels for our training participants and 37.6% improvement for our testing participants. In the Appendix, we provide graphical depictions of MIND MELD's ability to correct for suboptimal trajectories.

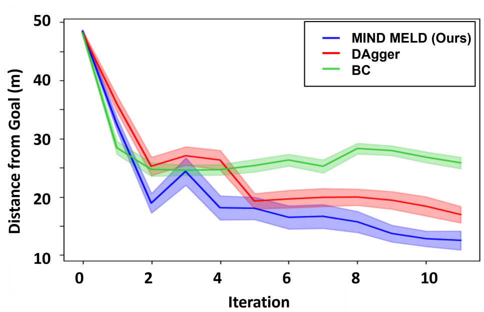


Fig. 5: This figure shows the average distance and standard deviation from the goal for each algorithm after each iteration. At each iteration, the agent rolls out the current policy and the participant provides a demonstration.

To test **Hypothesis 2**, we conduct a correlation analysis between the learned embeddings and the results of our dynamic time warping describing participants' over-/under-correcting and delayed/anticipatory tendencies. We find support for the results in our pilot study and find that the learned embeddings significantly correlate with participants' tendency to over-/under-correct ($r(116) = -.47, p < .001$) and provide anticipatory/delayed feedback ($r(116) = .49, p < .001$). To further investigate **Hypothesis 2** and determine if prior experience correlates with the learned embeddings, we conduct a correlation analysis between experience with driving and experience with video games. We find that experience with video games significantly correlates with the learned embedding ($\rho = .19, p = .038$).

To investigate **Hypothesis 3**, we next analyze the ability of each agent to reach the goal, in terms of both probability and frequency, over the course of the study. To determine the

	MELD-DAgger	MELD-BC	DAgger-BC
Workload	-8.1 (2.8) $p = .005$	-10.0 (2.8) $p < .001$	-2.0 (2.9) $p = .87$
Likeability	1.1 (.25) $p = .004$	1.4 (.28) $p = .001$	.31 (.27) $p = .37$
Intelligence	1.2 (.32) $p = .008$	1.7 (.31) $p < .001$	.53 (.31) $p = .35$
Trust	0.80 (.16) $p < .001$	1.1 (.14) $p < .001$	0.32 (.14) $p = .192$
Distance	-4.5 (.88) $p < .001$	-7.7 (.80) $p < .001$	-3.2 (.82) $p = .01$

TABLE I: We report the means (standard deviations) of the difference between the agents and associated p-values for objective and subjective metrics.

probability of reaching the goal at each iteration, we conduct a survival analysis, a statistical technique commonly used in medical research to assess the expected time until an event takes place [13]. Survival analysis allows us to analyze data for which an event may never occur. For example, an agent may never reach the orange ball during the study, yet we can still include this data in our survival analysis as "censored" data. In our study, time corresponds to the number of demonstrations that the agent has experienced. An event occurs when the agent reaches the goal the specified number of times.

Fig. 4 shows the Kaplan-Meier curves for reaching the goal once, twice, three times, and four times. We find that MIND MELD is statistically significantly more likely to reach the goal once (log rank $p < .001$), twice (log rank $p < .001$), three times (log rank $p < .001$), and four times (log rank $p < .001$) throughout the course of the study compared to DAgger and BC. We find that MIND MELD has a 100% chance of reaching the goal once after the seventh iteration whereas the baselines never achieve 100% probability of reaching the goal even once. Likewise, we find that MIND MELD has a $> 80\%$ chance of reaching the goal three times after the ninth iteration whereas the baselines have a $< 50\%$ chance. This result supports **Hypothesis 3** and shows MIND MELD learns a better policy in terms of probability of reaching the goal.

We additionally apply a Poisson regression with a Tukey post hoc to determine if there is a statistically significant difference between the total number of times that each agent reaches the goal throughout the study. We find that MIND MELD reached the goal 2.1x more than DAgger ($p < .001$) and 2.6x more than BC ($p < .001$).

Next, we analyze the average distance from the goal across iterations for each algorithm. We conduct a repeated measures ANOVA with a Tukey post hoc comparing the distance to the goal for each condition. As shown in Table I, we find that MIND MELD achieved a statistically significantly lower average distance from the goal ($M = 20.4, SD = 5.58$) compared to DAgger ($M = 24.8, SD = 5.92, p < .001$) and BC ($M = 28.2, SD = 4.86, p < .001$). Fig. 5 shows the average distance to the goal for each trial and condition. Note that a trial ends after the agent either reaches the orange ball or crashes into an obstacle.

To determine if our findings support **Hypothesis 4**, we conduct a correlation analysis between the participants' stylistic tendencies and the average difference between MIND MELD and DAgger. We find that participants' delayed/anticipatory tendencies significantly correlate with MIND MELD's advantage over DAgger ($r(40) = .36$, $p = .017$), as shown in Fig. 6.

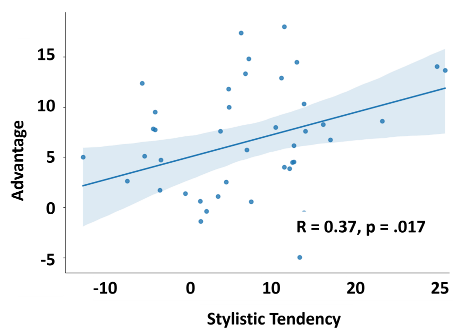


Fig. 6: This figure shows a plot of participants' tendency to provide delayed/anticipatory feedback vs. the difference between the average performance of MIND MELD and DAgger.

We lastly investigate our findings in the context of **Hypothesis 5** to determine if MIND MELD is rated subjectively higher by participants. We conducted a repeated measures ANOVA with a Tukey post hoc or Friedman's test (see omnibus statistics in the Appendix). As shown in Table I, MIND MELD is rated statistically significantly higher compared to both DAgger and BC for all subjective metrics. These findings support **Hypothesis 5**.

VII. Discussion

In our analysis, we find support for **Hypotheses 1-5**, illustrating that MIND MELD can learn stylistic tendencies of suboptimal and heterogeneous demonstrators, map the suboptimal feedback to better feedback, and, as a result, outperform prior work in both robot-centric and human-centric LfD. We find that MIND MELD is able to learn various participant styles, such as participants' tendency to over-/under-correct ($p < .001$) and provide delayed and anticipatory feedback ($p < .001$), suggesting that MIND MELD can provide positive results with a diverse user pool. For more discussion on stylistic tendencies, please refer to the Appendix.

Because MIND MELD is able to learn heterogeneous tendencies and utilize this information to correct for suboptimal behavior, we find that MIND MELD outperforms prior work in terms of its ability to reach the goal in an LfD task. MIND MELD achieves both a higher probability of reaching the goal and a lower average distance from the goal compared to both baselines, DAgger ($p < .001$) and BC ($p < .001$). Additionally, we observe that the more delayed a participant is at providing feedback, the better MIND MELD performs over DAgger ($p = .017$). We find that, for participants who provide less suboptimal feedback, MIND MELD and DAgger exhibit more similar performance because there is less of a need to correct a participants' feedback. When a participant's behavior deviates more from the optimal, DAgger performs worse, whereas MIND MELD is able to correct for the suboptimality.

Not only do we see improved performance in terms of objective metrics, we also find that MIND MELD outperforms both DAgger and BC in terms of our subjective metrics. Participants rate MIND MELD to be more likeable ($p = .004$), intelligent ($p = .008$), and trustworthy ($p = .001$) compared to DAgger. Additionally, we find the participants' perceived workload is rated as lower for MIND MELD ($p = .005$). This is an interesting finding considering that for both MIND MELD and DAgger, participants are tasked with providing corrective feedback to the agent. With respect to performance and human usability, MIND MELD achieves the best of both worlds. MIND MELD improves upon the performance of robot-centric algorithms, while being easy to teach, likeable, intelligent, and trustworthy.

VIII. Limitations/Future Work

Due, in part, to the recruiting difficulties imposed by the COVID-19 pandemic, our sample population consisted primarily of students with a mean age of 22.6. In the future, we plan to conduct this experiment with a more diverse set of participants. We also note that MIND MELD requires training participants to meta-learn the model parameters and a set of calibration tasks with ground-truth labels to learn the personalized embeddings. However, our results demonstrate that MIND MELD improves the quality of the corrective feedback by 37.6% and LfD outcomes ($p < .001$), making this additional step worthwhile.

Additionally, MIND MELD makes several assumptions, listed in Section III, about the way in which individuals provide corrective feedback. Yet, the success of our algorithm suggests that these assumptions appear to be sufficiently met for our experimental setup. For this study, we assume that a person's feedback style will remain constant; however, we do expect that, over a longer period of interaction, a person's style of feedback may change and adapt. In future work, we plan to investigate how to update our framework to account for and learn changing styles during longitudinal LfD.

Lastly, we aim to investigate if we can replicate the benefits of MIND MELD in other domains. We plan to implement MIND MELD on a robot arm domain, which may produce different behavior and stylistic tendencies amongst participants due to more degrees of freedom and a more complex user interface.

IX. Conclusion

We introduce MIND MELD, a novel LfD framework that learns personalized embeddings from heterogeneous users and improves upon suboptimal human feedback for robot-centric LfD algorithms. Through a human-subjects experiment, we showed that MIND MELD outperforms a human-centric baseline, BC, and a robot-centric baseline, DAgger, with regards to multiple measures of algorithm performance. Furthermore, users found MIND MELD more intelligent, likeable, trustworthy, and easier to teach than BC and DAgger.

REFERENCES

[1] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

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

[3] Christoph Bartneck, Elizabeth Croft, and Dana Kulic. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International Journal of Social Robotics*, 1(1):71–81, 2009.

[4] Jakob Berggren. Performance Evaluation of Imitation Learning Algorithms with Human Experts. 2019.

[5] W. Bradley Knox and Peter Stone. TAMER: Training an Agent Manually via Evaluative Reinforcement. In *2008 7th IEEE International Conference on Development and Learning*, pages 292–297, 2008.

[6] Daniel S. Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. 4 2019.

[7] Daniel S. Brown, Wonjoon Goo, and Scott Niekum. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. 7 2019.

[8] Eduardo F. Camacho and Carlos A. Bordons. *Model Predictive Control in the Process Industry*. Springer-Verlag, Berlin, Heidelberg, 1997.

[9] Letian Chen, Rohan R. Paleja, and Matthew Craig Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. In *CoRL*, 2020.

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems*, pages 2180–2188, 2016.

[11] Sonia Chernova and Andrea L. Thomaz. *Robot Learning from Human Teachers*. Morgan & Claypool Publishers, 2014.

[12] Sonia Chernova and Manuela Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34:1–25, 2009.

[13] William N Dudley, Rita Wickham, and Nicholas Coombs. An introduction to survival statistics: Kaplan-meier analysis. *Journal of the advanced practitioner in oncology*, 7(1):91—100, 2016.

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. 2017.

[15] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human mental workload*, 1(3):139–183, 1988.

[16] He He, Hal Daumé, and Jason Eisner. Imitation Learning by Coaching. *Conference on Neural Information Processing Systems*, pages 1–9, 2012.

[17] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning.

[18] Rohit Jena, Changliu Liu, and Katia Sycara. Augmenting GAIL with BC for sample efficient imitation learning. pages 1–11, 2020.

[19] Jiun-Yin Jian, Ann Bisantz, and Colin Drury. Foundations for an Empirically Determined Scale of Trust in Automated Systems. *International Journal of Cognitive Ergonomics*, 4:53–71, 2000.

[20] Sertac Karaman and Emilio Frazzoli. Incremental sampling-based algorithms for optimal motion planning, 2010.

[21] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:8077–8083, 2019.

[22] Michael Laskey, Caleb Chuck, Jonathan Lee, Jeffrey Mahler, Sanjay Krishnan, Kevin Jamieson, Anca Dragan, and Ken Goldberg. Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 358–365, 2017.

[23] Michael Laskey, Sam Staszak, Wesley Yu Shu Hsieh, Jeffrey Mahler, Florian T. Pokorny, Anca D. Dragan, and Ken Goldberg. SHIV: Reducing supervisor burden in DAgger using support vectors for efficient learning from demonstrations in high dimensional state spaces. *Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June:462–469, 2016.

[24] Ruisen Liu, Matthew C Gombolay, and Stephen Balakirsky. Torwards Unpaired Human-to-Robot Demonstration Translation Learning Novel Tasks. *ICSR Workshop Human Robot Interaction for Space Robotics (HRI-SR)*, 2021.

[25] Kunal Menda, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. EnsembleDAgger: A Bayesian Approach to Safe Imitation Learning. *IEEE International Conference on Intelligent Robots and Systems*, (2):5041–5048, 2019.

[26] Takayuki Osa, Gerhard Neumann, and Jan Peters. An Algorithmic Perspective on Imitation Learning. 7(1):1–179, 2018.

[27] Brandon Packard and Santiago Onta. A User Study on Learning from Human Demonstration. (Aiide):208–214, 2018.

[28] Rohan Paleja and Matthew Gombolay. Inferring personalized bayesian embeddings for learning from heterogeneous demonstration. *arXiv*, 2019.

[29] Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. *Recent Advances in Robot Learning from Demonstration*, volume 3. 2020.

[30] Laurel D. Riek. Wizard of oz studies in hri: A systematic review and new reporting guidelines. *J. Hum.-Robot Interact.*, 1(1):119–136, July 2012.

[31] Stéphane Ross and J. Andrew Bagnell. Efficient reductions for imitation learning. *Journal of Machine Learning Research*, 9:661–668, 2010.

[32] Stéphane Ross, Geoffrey J Gordon, and J. Andrew Bagnell. No-regret reductions for imitation learning and structured prediction. *Aistats*, 15:627–635, 2011.

[33] Stéphane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J. Andrew Bagnell, and Martial Hebert. Learning monocular reactive UAV control in cluttered natural environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 1765–1772, 2013.

[34] S. Salvador and P. Chan. FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. 2004.

[35] Claude Sammut. Automatically Constructing Control Systems by Observing Human Behaviour. *Second International Inductive Logic Programming Workshop*, (May), 1992.

[36] Stefan Schaal. Learning from demonstration. In M. C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1997.

[37] Mariah L. Schrum, Erin Hedlund, and Matthew C. Gombolay. Improving Robot-Centric Learning from Demonstration via Personalized Embeddings, 2021. _eprint: 2110.03134.

[38] Mariah L. Schrum, Michael Johnson, Muyleng Ghuy, and Matthew C. Gombolay. Four years in review: Statistical practices of likert scales in human-robot interaction studies. *ACM/IEEE International Conference on Human-Robot Interaction*, pages 43–52, 2020.

[39] Aran Sena and Matthew Howard. Quantifying teaching behavior in robot learning from demonstration. 2020.

[40] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles, 2017.

[41] Jarrod M Snider. Automatic steering methods for autonomous automobile path tracking, 2009.

[42] Jonathan Spencer, Sanjiban Choudhury, Matt Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Siddhartha Srinivasa. Learning from Interventions: Human-robot interaction as both explicit and implicit feedback. 2020.