Interactive Robot Task Learning: Human Teaching Proficiency with Different Feedback Approaches

Lukas Hindemith, Oleksandra Bruns, Arthur Maximilian Noller, Nikolas Hemion, Sebastian Schneider and Anna-Lisa Vollmer

***Abstract*—The deployment of versatile robot systems in diverse environments requires intuitive approaches for humans to flexibly teach them new skills. In our present work, we investigate different user feedback types to teach a real robot a new movement skill. We compare feedback as star ratings on an *absolute scale* for single roll-outs versus *preference-based* feedback for pairwise comparisons with respective optimization algorithms (i.e., a variation of co-variance matrix adaptation - evolution strategy (CMA-ES) and random optimization) to teach the robot the game of skill cup-and-ball. In an experimental investigation with users, we investigated the influence of the feedback type on the user experience of interacting with the different interfaces and the performance of the learning systems. While there is no significant difference for the subjective user experience between the conditions, there is a significant difference in learning performance. The *preference-based* system learned the task quicker, but this did not influence the users' evaluation of it. In a follow-up study, we confirmed that the difference in learning performance indeed can be attributed to the human users' performance.**

***Index Terms*—Human-Robot Interaction Study, User Study, Human-in-the-loop robot learning, Interactive Machine Learning, Human Feedback, Radial Basis Function Network, Programming by Demonstration, Learning from Demonstration, Preference Learning.**

I. Introduction

FLEXIBLE industrial robots interacting with people, as well as service robots assisting people in need in domestic environments, are two domains where researchers, entrepreneurs, and policymakers expect robots to enter in our everyday lives soon [1]. To make robots versatile and flexible for varying scenarios and new tasks, people need to be able to teach them different behaviors. Therefore, one requirement is that the interfaces for the internal learning mechanisms are intuitively usable for everyone. A common approach to solve this problem utilizes *Programming by Demonstration* (PbD), where users show the robot how to do a movement (e.g., kinesthetic teaching) [2]. The robot can then reproduce the trajectory and, due to the imprecision of the demonstration or its own imitation (influenced by sensors for recording the demonstrated trajectory, the representation of the movement, possibly mapping it onto its own body, and control and hardware), optimize the final task performance by self-improvement using a pre-defined cost function. Since, designing the cost function is one of the bottlenecks, even for experts, it is unrealistic that non-expert users could develop such a function to teach the robot a new skill. Moreover, the cost is computed using measurements on the performance (e.g., via external camera setups) which is not suitable for domestic environments. Therefore, this work investigates the applicability of an optimization system from a user-centered perspective and investigates what kind of user feedback for the optimization is intuitively usable without much effort. We base our work on recent work by Vollmer and Hemion [3], who have shown that naive users can teach robots complex continuous movement skills via a simple user interface. We here also concentrate on robot learning for complex movement skills with a human teacher and compare the types of feedback a teacher could give as a performance measure: feedback as star ratings on an *absolute scale* for single roll-outs (as in [3]) versus *preference-based* feedback for pairwise comparisons. In the following we will refer to the two conditions as 'absolute scale' and 'preference-based', respectively.

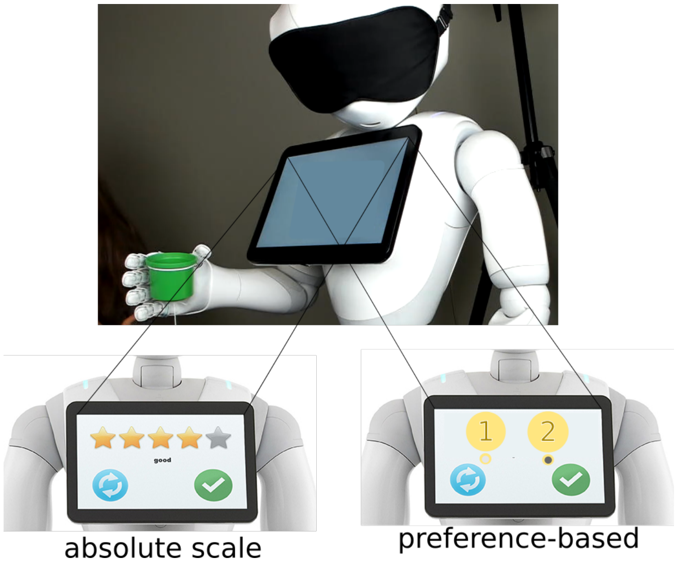


Fig. 1: User Interfaces used in this experiment.

Investigating feedback types for robot learning is fundamental because they introduce advantages and disadvantages. On the one hand, absolute scale user feedback can more easily be transformed into a reward signal for the learner. However, asking people to rate movements on an absolute scale comes with some drawbacks [4]: 1) scales vary between different

L. Hindemith, A. M. Noller and A.-L. Vollmer are with the Medical School OWL and Center for Cognitive Interaction Technology, Bielefeld University, Germany e-mail: anna-lisa.vollmer@uni-bielefeld.de).

O. Bruns is with IZ Karlsruhe – Leibniz Institute for Information Infrastructure and Karlsruhe Institute of Technology, Germany.

N. Hemion is with dSPACE GmbH, Paderborn, Germany.

S.Schneider is with Institut für Produktentwicklung und Konstruktionstechnik, TH Köln, Germany.

Manuscript received XXX; revised XXX.

users, 2) human evaluation is influenced by *anchoring*, where early experience dominates the scale [5], and 3) evaluation is also subject to drift, where scales change over time [cf. 6]. This is supported by the number of users who were not able to successfully teach the robot, because their strategy was not compatible with the properties of the underlying learning algorithm in [3]. On the other hand, outside of interactive task learning, people have been shown to be very proficient at giving preference-based feedback and at comparing things [7].

We thus hypothesize that users are more proficient in teaching a robot a new movement skill when giving preference-based feedback than when giving feedback on an absolute scale. On that account, their subjective experience of the teaching interaction should be more positive when teaching with preference-based feedback. This condition should be more positively evaluated with respect to: 1) the task success 2) their satisfaction with the task 3) their perception of how social the robot is 4) the system usability 5) their task load Consequently, from the higher proficiency of users in teaching with preference-based feedback should additionally result a higher performance of the learning system in the preference-based feedback condition. From a user-centered perspective, this work tackles the research questions of how teaching interfaces for robot learning in interaction should be designed and how humans can actively shape a machine learning process as teachers.

A. Related Work

This work combines two active research fields: Interactive Machine Learning (IML) and Human-Robot Interaction (HRI). It targets the questions how humans can actively shape a machine learning process as teachers and how the teaching interfaces for the learner, that have to tightly couple front-end and back-end, should be designed.

Prior work studying how humans can provide reward signals for reinforcement learning algorithms exists [8, 9, 10, 11], where most of the studies in the literature consider discrete robot actions. Thus, basic actions of solving a task were known a priori. For example, Thomaz et al. [9] presented work on how to use user input as a reward signal for a reinforcement learning agent that learns a sequential task in a virtual environment, and how the algorithm needs to be altered according to the results from their user studies. This work was followed by work from Senft et al. [12] who also researched how people use numeric feedback to guide the robot learning. The limitation of the above mentioned works is that the discrete action space limits the possible actions one can teach the robot. In our case, we are interested in a continuous action space that allows to teach a robot new actions or building blocks for new skills. For related work in the area of user feedback for continuous action spaces see [13, 14].

Other works, not relying on a user generated absolute scale reward signal, suggest to use preference-based learning [15]. In these learning scenarios, users are iteratively presented two alternative behaviors from an (robotic) agent and asked to give a preference statement, selecting one behavior over the other. Sadigh et al. [16] showed that by using their approach people could teach a simulated 2D autonomous car to drive in a way users found reasonable. One drawback of their approach is that the system used predefined feature representations for the cost function estimation which is similarly challenging as designing a cost function. Additionally, preference-based learning was also utilized in a deep reinforcement learning task, where users watched pairs of short video clips showing a virtual agent's behavior (e.g., simulated robots, Atari games) and could give feedback according to their preference [17]. This approach let the agent learn complex behaviors and reduced the time humans had to teach the learning system. Since this work uses deep reinforcement learning to teach the agent, it requires hundreds of hours to train the agent, which is a limitation for using it on a physical robot due to the time necessary for training and the fact that the hardware will wear down quickly. Therefore, we present a system that does not need a predefined cost function or feature representation, and can learn successful movement skills from non-expert users in a couple of minutes. In contrast to work done by Vollmer and Hemion [3], we have looked into the drawbacks of using *absolute-scale* feedback from users, which is influenced by a drift in evaluation and the requirement of anchoring to a reference point, by utilizing *preference-based* feedback from users.

Since it is challenging to anticipate how non-expert users will behave, it is important to early on look at the human factors that are important when teaching machine learning systems. For example, Cakmak and Thomaz [10] have shown that human teachers might not use the optimal strategy to teach such a system. Thus, it is important to provide interfaces to the users that guide them to use an optimal teaching strategy. Therefore, in our present work, we focus on how to constrain the possible user feedback so that the machine learner receives optimal user feedback. Additionally, in contrast to work focusing on how naive users teach a robot a new skill via kinesthetic teaching (e.g., [18]), we are not looking at the demonstration part of the skill acquisition, but on the effectiveness of different user feedback approaches as a replacement for a cost function.

B. Contribution and Outline

In this work, we compare naive, *absolute scale* user feedback for a black box optimization algorithm against naive, *preference-based* user feedback for a random optimization variant (Study 1). In a follow-up study, we then compare the two algorithms with objective feedback obtained as distances from ball to cup using a camera setup to validate that differences in performance are due to the proficiency of the humans rather than one optimization algorithm being objectively superior over the other (Study 2).

The paper is outlined as follows: The next section presents Study 1. It first gives an overview of the system design including the robot, the study setup, the learning algorithm and the user interface. Section II-B summarizes our study design to evaluate the different feedback approaches. The results of our study are highlighted in Section II-C and discussed in Section

II-D. Section III is structured into the same subsections and presents the camera-based optimization.

II. Study 1: Human Feedback

A. System Design

1) Robot: For the present work, we used a Pepper humanoid robot platform developed and sold by SoftBank Robotics. Pepper is 1.2 m tall and its design is intended to foster natural and intuitive interactions with humans. A 10.1-inch tablet is attached to the front of its torso and here functions as an input device. Pepper's operating system is NAOqi OS. In our study, we only used Pepper's right arm for movements, the left arm and the body were in a fixed position. We disabled the collision avoidance of the robot and set its joint stiffness to 70% to prevent frequent overheating.

2) Cup-and-Ball: The cup-and-ball game is a children's game of skill. The toy is a cup (with a diameter of $2.2in$ and a height of $2.2in$) with a ball (with a diameter of $1in$) attached to it with a string (of $9in$ length). The goal of the game is to catch the ball with the cup through skillful movement. Kober and Peters [19] have demonstrated that the cup-and-ball movement can be learned by a robot arm using DMP-based optimization, and Vollmer and Hemion [3] have demonstrated that Pepper is capable of mastering the game with human absolute scale ratings as reward. In this study, the cup-and-ball toy was built such that the size of the cup and ball resulted in a level of difficulty suitable for our purposes. This means that it resulted in an agreeable user experience by a minimized time spent on *fine tuning* the movement near the cup at the end of the optimization and by a minimized time necessary to teach the skill until it has been successfully learned.

3) Learning algorithm and User Interface:

a) Movement representation: As we use only the right arm of the robot, all together five joints are employed for the movement execution (three in the shoulder, one in the elbow, one in the wrist). The robot movement execution thus corresponds to a sequence of arm configurations $(\mathbf{x}_t)_{t=1}^T = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, with $\mathbf{x} \in \mathbb{R}^5$, sampled at 60 Hz. Optimizing the trajectories directly is infeasible within the scope of an interaction[1]. Instead, we use function approximation to reduce the number of parameters to be optimized. We approximate the sequence $(\mathbf{x}_t)_{t=1}^T$ by means of a Radial Basis Function Network (RBFN) that takes as input a scalar representation of time, $t \in [0, T]$, s.t.

$$\mathbf{x}_t = \mathbf{x}_0 + f(t, W) = \mathbf{x}_0 + W \cdot \mathbf{h}(t). \qquad (1)$$

where $\mathbf{h}(t) = (h_1(t), \ldots, h_n(t))$ is a vector of $n = 10$ fixed radial basis functions evaluated at t, and $\mathbf{x}_0$ is the arm configuration at the beginning of the movement. This way, we no longer optimize all x_t directly, but instead optimize the weights W of the RBFN (50 parameters in total). As basis functions, we use a set of Gaussian functions (see Appendix A for details).

b) Optimization: For learning the cup-and-ball skill, we use a PbD approach as explained in Section I. The initial demonstration from which the system optimized the movement was provided to the system by an experimenter via kinesthetic teaching. All subjects in both conditions started from this first demonstration which fell short of successfully hitting the cup with the ball.

Participants either gave absolute scale or preference-based reward for which two different optimization approaches were used. They are described in the following. The main difference between approaches is that updates in the absolute scale feedback condition happen batch-wise – after ten demonstrations, whereas in the preference-based feedback condition, parameters are updated after each compaired pair of roll-outs. The underlying representation in both approaches (i.e., DMP) is identical. To ensure comparability of the two optimization approaches, we thus set the parameters of the decay factor for exploration to result in the same convergence rate per generated sample. All parameters and their values are listed in Table I.

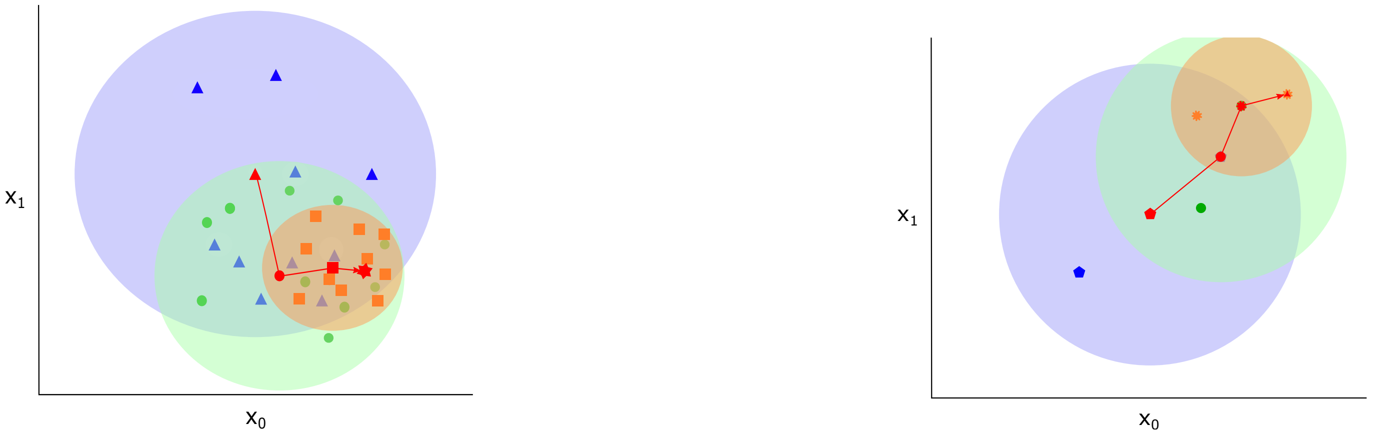
TABLE I: Overview of the open parameters of the system which influence learning.

Parameter	Value for absolute scale	Value for preference-based
Initialization	Same for both conditions.	
Stiffness	70 %	70 %
# basis functions	10	10
Exploration rate	0.01	0.01
Batch size	10	n/a
Decay factor	$\lambda_1 = 0.77378$, s.t. $(\lambda_1)^8 = c$	$\lambda_2 = 0.95$, s.t. $(\lambda_2)^{40} = c$

c) Absolute Scale Reward: For optimization with costs from discrete, absolute scale rewards, we use simple black-box optimization for updating the parameters [20]. We use the Path Integral –Black Box Optimization (PIBB) algorithm which functions similar to gradient descent. At each iteration, a batch of 10 samples from a normal distribution with covariance (exploration rate in each entry updated with the decay factor) around the mean is generated (see figure 2a). Each sample in a batch is performed by the robot separately and the user gives each sample movement a scalar rating on a scale from 1 to 5. The user interface for rating shows five stars as for common product reviews, a button with which the current sample can be replayed, and a button for confirming the rating (see interface on the left of Fig. 1). Once all ten ratings for a batch have been received, the new mean for the next iteration is computed via reward-weighed averaging.

d) Preference-based Reward: For learning the Cup-and-Ball skill from preference-based rewards, we use a simple algorithm similar to random optimization [21]. At each iteration, two samples are generated. A multivariate Gaussian random vector is generated and added to and subtracted from the current mean respectively. The two samples thus lie in opposite directions from the mean (see Fig. 2b). The two samples are presented to the user subsequently (i.e., the robot performs the two movements) and the user is asked to choose the one they

[1] as for example for a movement duration of 5 seconds, a trajectory is represented as a $(5 \cdot T) = 1500$-dimensional vector



(a) PIBB sampling with a batch size of 10.

(b) Random Optimization Variant.

Fig. 2: Schematic visualization of PIBB sampling (left) and the Random Optimization sampling (right). Both schemes show the optimization for 3 epochs in a 2-dimensional representation space. Blue, green and orange symbols depict the samples of first, second and third epoch respectively. The samples lie in the exploration radius, marked as colored ellipses around the current mean for each epoch. Red symbols show the means of the batches. Red arrows indicate update steps.

prefer. The user interface is designed as a website with buttons for each sample and a button for choice confirmation (see Fig. 1). Parameters are updated with each user rating. The sample preferred by the user has the lowest cost and is chosen as the new mean for the next iteration.

B. Study Design

In the following, we will describe the conducted study with non-expert users that have little experience with robots. It was conducted at Bielefeld University and approved by the local ethics committee. We obtained written informed consent from all participants. We conducted a between-subject design, where participants interacted with a physical robot in our lab. We chose a between-subject design to prevent spill-over effects concerning teaching strategy. Participants were randomly assigned to one of the following conditions:

a) absolute scale: In the absolute scale feedback condition, participants saw one roll-out (i.e., a single movement of the robot) and could rate the performance of the roll-out using a 5-point Likert-scales (1: not good at all, 2: not so good, 3: average, 4: good, 5: very good).

b) preference-based: In the preference-based condition, the participant was shown two roll-outs of the robot and could give a preference-based feedback on which one was better.

1) Participants: Participants were recruited through flyers and advertisements on campus and on social media. 30 participants (10 m, 18 f, 1 d, 1 N/A age: M = 25.37, SD = 7.39) took part in the experiment.

2) Experimental Setups: The experiment took place in a laboratory at Bielefeld University. The participant was sitting in front of Pepper. The experimenter sat to the left of the participant (see Fig.3).

3) Course of the experiment: The course of the experiment can be seen in Fig. 4. First, the experimenter instructed the participant (in German) that the research conducted is about robot learning and that the study they participate in tests the learning of the robot Pepper and if humans are able to teach it the game cup-and-ball. The goal of the task is that Pepper gets the ball into the cup by moving its arm. Pepper is blindfolded when learning the task. The cup is attached to Pepper's hand and the experimenter will assure that the ball is hanging still from the cup before movements. Thus, participants had no information about the learning algorithm. We also did not show them what the movement for the game looked like in order to avoid priming them about possible task performance. For the two conditions, we had the following instructions:

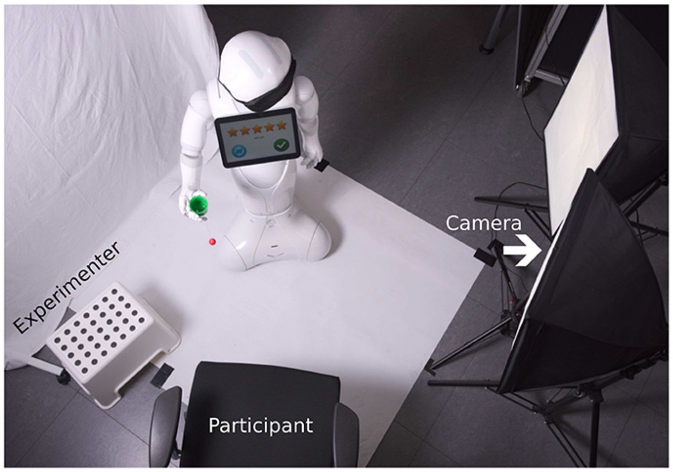


Fig. 3: Experimental setup from above. Image taken from [3]

a) absolute scale condition: The participant was told that a GUI is displayed on the robot's tablet showing five stars as for common product reviews with which they could rate each movement (see Fig. 1). The stars correspond to the ratings of (common 5-point Likert-scales) 1: not good at all, 2: not so good, 3: average, 4: good, 5: very good. Ratings were confirmed via a green check mark button on the lower right of the GUI. A movement could be repeated via a replay button on the lower left. Upon confirmation of a rating, the score was transformed into a cost by inverting the scale. The tablet then showed a ready prompt to give the experimenter enough time to reposition the ball such that it was hanging straight without swinging. After another confirmation on the GUI, the robot directly showed the next roll-out.

b) preference-based condition: The participant was instructed that they are shown two movements by pressing the movement buttons *1* and *2* respectively on a GUI shown on the robot's tablet (see interface on the right in Fig. 1). Following, a movement could be selected via small radio buttons below the movement buttons, indicating their feedback on which roll-out, *1* or *2* they preferred. A preference feedback is confirmed via the green check mark button on the lower right of the screen. Participants could see a movement again, if needed, by pressing the respective movement button again. When the rating was confirmed, the parameters of the associated movement of the roll-out were chosen as the new mean for the next iteration. A ready prompt screen was then shown to allow the repositioning of the ball to hang still from the cup.

After the instructions, Pepper introduced itself with its standard autonomous life behavior, looking at the participant and gesturing. Pepper said that it wanted to learn the game while being blindfolded but did not know yet how exactly the game works. It further explained that it would try several times and the participant had to help by telling it how good each roll-out was (in the absolute scale condition) or which roll-out was better (in the preference-based condition).

The participant then started to rate the roll-outs Pepper showed to them. The maximum number of roll-outs was 80, but we defined an additional abort criterion. When Pepper performed five hits in a row (i.e., five consecutive movements that landed the ball in the cup), irrespective of batches or pairs of movements, we defined learning as being successful and aborted the experiment with the fifth hit. At the end of the learning, Pepper moved into its resting position thanking the participant and explaining that it now needed some rest. At this point, the experimenter took over and asked participants to fill out an online questionnaire (for details please see Section II-B4). When the participant was finished, the experimenter conducted a structured interview on participants' individual strategies.

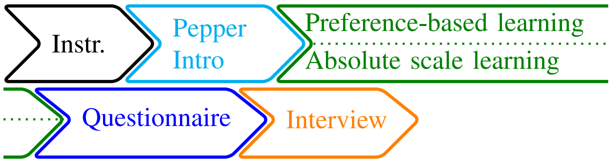


Fig. 4: Course of the experiment in Study 1.

4) Measurements: To measure the difference in terms of the user experience and to evaluate the learning performance, we used post-study surveys, quantitative evaluation measures for performance and structured interviews.

a) Survey Questions: To compare the participants' subjective impressions of the feedback type and interaction behavior, we used five different established questionnaires that measure the task enjoyment, the perception of the robot, usability and task load.

Task Success: We asked the participants how successful the robot was in learning the task on a 5-point Likert scale with four items. (e.g., 1: *not successful* – 5: *very successful*).

Task Satisfaction: We asked participants how much they enjoyed the interaction on a 5-point differential scale (e.g., *I enjoyed it* – e.g., *I did not enjoy it*) with 16 items.

Robotic Social Attribute Scale [22]: We used the Robotic Social Attribute Scale (RoSAs), as a measurement for the participants' perception of the robot regarding its competence, warmth and whether they felt discomfort on a 9-point Likert scale (1: 'I definitely not associate the attribute with the robot'; 9: 'I definitely associate the attribute with the robot').

System Usability Scale [23]: We used the System Usability Scale (SUS) as a measurement for the usability of the feedback type on a 5-point Likert scale (e.g., *I experience the system as unnecessary complex*, 1: 'I totally agree'; 5: 'I totally disagree').

Task Load Index [24]: To measure the participants task load, we used the NASA Task Load Index (TLX) on a 5-point Likert scale (e.g., *How hard did you have to work, to achieve your level of task success?*, 1: 'low'; 5: 'high').

b) Quantitative Measurements: Additional to the subjective survey measurements, we used objective quantitative measurements to assess the effectiveness of the different learning strategies.

Interaction Time: We measured the total interaction time in seconds from the moment when the participants started to teach the robot, until we reached the abort criterion. The abort criterion for the conditions are explained in Section II-B3.

First Hit: The first hit is defined as the number of the roll-out when the ball landed in the cup for the first time.

Hit Ratio: We measured the ratio between the number of hits (i.e., the ball landed in the cup) and the total number of roll-outs until we meet the abort criterion.

$$hitratio = \frac{\#hits}{total\#roll-outs} \tag{2}$$

where the total number of roll-outs is either 80, if the system does not have five consecutive hits, otherwise it is the number of roll-outs until five consecutive hits have been performed.

c) Qualitative: Additionally, we conducted structured post-study interviews to gain insights into participants' cognitive strategies to evaluate the performance of the robot, and whether they changed their strategy. We asked the participants

- What is your criterion for successful learning?
- Have you evaluated the performance spontaneously or strategically?
- Do you think you made mistakes in the evaluation? If yes, how?
- What was your strategy for evaluating the robot behavior?
- Did you change your strategy? Consciously?
- Have you motivated the robot? If yes, how?

C. Results

The results were analyzed using the statistical computing language R [25]. To compare the different conditions, we use the non-parametric Wilcoxon rank sum test, due to the small number of test samples [26]. In the following, we describe the results from the survey responses, the quantitative results from the teaching success as well as the qualitative interview responses.

1) Survey Questions: The mean, standard deviation, and statistical results for the survey responses are listed in Table II. The results show no differences between the conditions in any measurements. Participants in both conditions did not evaluate the task load, the perception of the robot, the task success, or the usability of the system significantly differently.

2) Quantitative Results: The quantitative results of *first hit* and *hit ratio* are depicted in Figure 5. The system with the *preference-based* feedback resulted in a significantly earlier first hit (M = 18.93, SD = 21.41) compared to the *absolute scale* feedback system (M = 36, SD = 20.69), W = 187.5, $p <$ .01. Additionally, the overall ratio of hits to total roll-outs was significantly higher for the *preference-based* condition (M = .3, SD = .17) compared to the *absolute scale* condition (M = .18, SD = .13), W = 62, $p <$ .05. The total interaction time between the conditions was not significantly different, W = 79, p = .58. Participants in the *preference-based* condition spent on average M = 432.3 seconds (SD = 153.98) teaching the robot the skill and participants in the *absolute scale* condition needed M = 388.8 (SD = 72.85) seconds for the task.

3) Interview: Part of the interview responses are listed in Table IV and frequency counts are listed in Table III. Interview results show a significant difference for the *strategy changed* ratio between the conditions, χ^2 = 15.52, p $<$ 0.01.

a) Strategy: Most participants in the *preference-based* condition used a strategy based on the distance to the cup ($n = 12$). One participant stated that they used an intuitive strategy and two participants additionally stated that they used the momentum as a strategy. Participants in the *absolute scale* condition also used the distance to the cup as an evaluation criterion but changed their evaluation criterion over time. In the beginning of the learning phase, participants gave high ratings when the ball was close to the rim of the cup. After the first successful roll-out, participants adapted their strategy and only gave high feedback when the ball landed in the cup ($n = 10$). Only one participant stated that they gave the same ratings during the whole experiment.

b) Errors: Nine participants in the *absolute scale* condition said that they made errors during the rating. In the *preference-based* condition, ten participants said that they made errors. Participants in the *absolute scale* condition explained that they did not know which errors they made or that they were unsure in the beginning. While participants in the *preference-based* condition said that the movements were too similar ($n = 3$), and that they were unsure and it was difficult to estimate ($n = 6$).

c) Motivating: In the *absolute scale* condition, two participants said that they tried to motivate the robot by giving it better ratings.

D. Discussion

In this work, we have presented two kinds of teaching strategies for giving robots feedback on learning a movement skill. We have implemented two distinct but comparable feedback algorithms that either rely on the given absolute scale reward or preference-based comparative feedback from a human teacher. To investigate the user experience and performance differences between these feedback mechanisms, we have conducted a between-subject design study in which participants taught the robot how to play the game of skill *cup-and-ball*.

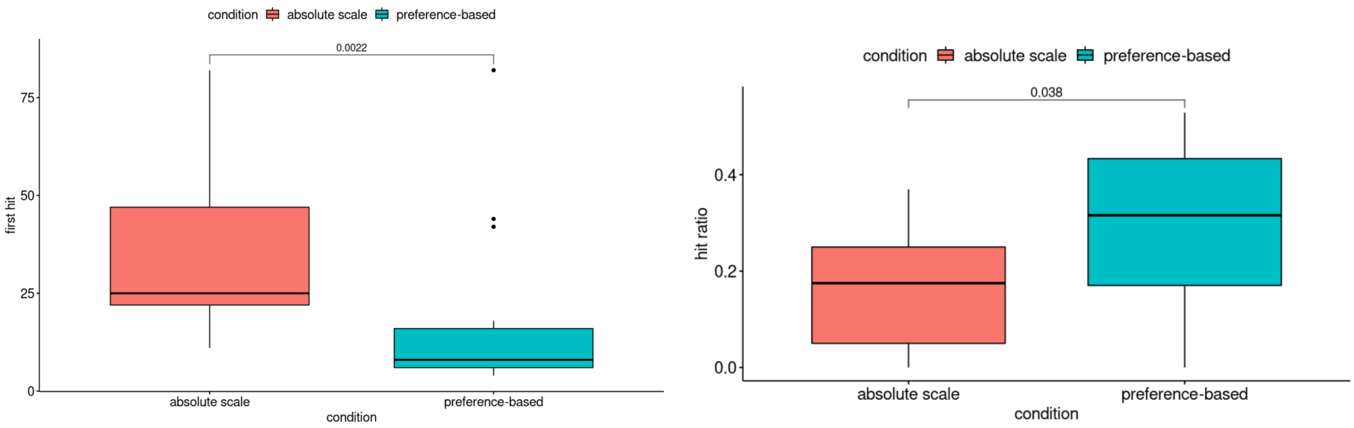
Our results presented in the previous section show that there is no significant difference between the conditions regarding the evaluation of the task or the robot. **We were thus unable to confirm our hypothesis on a more positive subjective user experience when teaching with preference-based feedback.** Both cohorts equally perceived the task as successful, satisfying, usable, and demanding. Additionally, the ratings for perceived competence, warmth, or discomfort regarding the robot were not significantly different between the groups. The results show that participants rated the task success high in both conditions and showed a *stick to the middle* response for task satisfaction. This is no surprising result because, in most cases, the robot was able to learn the task successfully. For three users in the *preference-based* and eight users in the *absolute scale* condition the abort criterion was not met, however, for only one user in each condition, the system was not able to learn the skill. In a similar vein, the high competence evaluation in both conditions might not be due to the different feedback types, but because in both conditions, participants saw a robot for the first time, which was doing a skillful task. The ratings on the warmth scale are probably also similar because the robot looked the same and said the same things in the experiments. Additionally, the low task load in both conditions might be explained by the phenomenon of giving socially desirable responses at surveys. Participants might not want to admit that the task was demanding.

The results mentioned above relate to the first question our work is concerned with: how should a teaching interface for the learner that has to tightly couple front-end and back-end be designed? Our results give no conclusive answer. Neither of the interfaces changed the user's subjective impression, and there is no favor for one or the other. However, the interview responses highlight that the users used different strategies. In the absolute-scale case, they developed an explicit heuristic to evaluate the performance numerically. In contrast, in the preference-based case, the users relied on an intuitive evaluation method.

Even though we found no subjective difference between

TABLE II: Mean [M], standard deviation [SD] and Wilcoxon rank sum test (WRST) statistics for survey result.

Measurement	*absolute scale* M (SD)	*preference* M (SD)	WRST statistics
Task Success	4.18 (.78)	3.73 (.85)	W = 139, p = .14
Task Satisfaction	3.62 (.56)	3.57 (.56)	W = 119.5, p = .54
TLX	1.91 (.47)	2.06 (.75)	W = 95.5, p = .7
warmth	3.78 (1.58)	3.27 (1.7)	W = 127, p = .34
competence	6.43 (1.15)	5.87 (1.71)	W = 123.5, p = .43
discomfort	2.61 (1.73)	2.96 (2.06)	W = 103, p = .95
SUS	2.09 (.55)	2.24 (.88)	W = 102.5, p = .93



(a) Number of roll-outs needed for the first hit.

(b) Number of hits to total number of roll-outs ratio.

Fig. 5: Box plot results for Study 1: user feedback.

TABLE III: Frequency table of interview results.

condition	strategy changed	strategy not changed
absolute scale	6	9
preference-based	0	14
	spontaneous	not spontaneous
absolute scale	3	11
preference-based	3	12

the conditions, the *preference-based* system performed better. **This finding confirms our second hypothesis that the performance of the learning system is better in the preference-based condition** and relates to our second research question on how humans can actively shape a machine learning process as teachers. The *preference-based* skill teaching led to a significantly earlier first hit and it also produced substantially more hits than the *absolute scale* based system. Eventually, the comparative feedback between two options led to a more careful evaluation of the robot's performance, due to the possibility of having an anchor to compare the performance. An investigation of the participants' behavior showed that they used the replay function of the system more often in the *preference-based* condition (M = 1.14, SD = .13) than in the *absolute scale* condition (M = 1.02, SD = .03), , W = 55.5, $p <$ .05. Arguably, the higher use of the replay function could attribute to the found differences in terms of convergence speed. Our analyses here were inconclusive: we were unable to find a significant correlation between the number of replays per roll-out and, respectively, the first hit and hit ratio variables, however this should be confirmed in a follow-up study with a larger number of participants.

Surprisingly, the number of samples until a first hit did not lead to a difference in the evaluation of the system's competence or usability. This non-existing difference might be because, in most cases, the system did learn the skill. Nevertheless, our results are in line with other works that showed the advantage of using *preference-based* teaching approaches and that humans are better in giving comparative feedback [7].

Moreover, the user's comprehension of the underlying functionality of the algorithm needs to be considered. A user's understanding of how samples are utilized to update the means of the RBFN affects the performance. As the preference-based method directly utilizes the user-selected roll-out for the update, this method is closely related to users' expectations of the system. In contrast, the absolute-scale method updates the weights after all roll-outs of a batch. This is information users were not aware of. As already observed in Vollmer and Hemion [3], users apply different strategies when teaching with this algorithm. While strategies, such as the distance from the ball to the cup or the momentum are appropriate, some users apply comparative or spontaneous ratings. Consequently, the possible mismatch between users' mental model of a learning algorithm and its true functionality demands new strategies and aspects of algorithm design. With the goal of interactive task learning, algorithms need to be designed in a manner users can comprehend. Thereby, users apply suitable strategies for teaching the robot.

Additionally, the performance difference between the conditions in Study 1 cannot with certainty be explained alone by the qualitatively better feedback from the human teacher. Possibly, the *preference-based* feedback algorithm performs better, even without a human. To test this, we conducted a follow-up study (Study 2) with no human teacher involved but with a feedback signal coming from an external camera setup that uses the distance to the cup as an objective measurement to compare which of the two approaches performs better objectively. The results from this study help to evaluate whether the performance differences are due to the algorithm or due to better human feedback.

III. Study 2: Camera-based Optimization

In Study 2, we evaluated the two optimization approaches with the same task and objective costs. If in this study the *absolute scale* approach performs the same or even better than the *preference-based* approach, we can attribute the results of Study 1 to the human feedback.

TABLE IV: Participants' responses for the interview questions on the used strategy for giving feedback and whether they made errors when giving feedback (as: absolute-scale; pb: preference-based).

condition	id	described strategy	errors
as	1	first spontaneous, when it got closer to the cup my evaluation was more strict	–
	2	five stars for touching the cup, later only for hits	I was not sure at the beginning
	3	first spontaneous	–
	5	–	Overrated
	6	if it was slight improvement, I would vie more star. I was comparing the moves	I don't know
	7	3 for rim, 4 sloppy hit, 5 clear hit	–
	8	First 5 stars close to cup, later 5 stars only for hits	–
	10	First 5 stars for hits, 4 for rim, 1 weak movement	I was not sure
	11	First 5 stars close to cup, later 5 stars only for hits	–
	12	momentum of movement and distance ball to cup	I made subjective errors
	13	I wanted that it changed its movement when I gave strict evaluations, but that did not happen	I was too strict
	14	distance ball to cup	I made subjective errors
	15		I don't know
	16	First 5 stars close to cup, later 5 stars only for hits	I don't know
pb	1		I did not understand how it works
	2	intuitive	–
	3,9	momentum of movement and distance ball to cup	Not sure
	5	success, distance ball to cup	I was unsure
	6-8;10-15	distance ball to cup	movements were similar

A. System Design

The underlying system was the same as for the previous study. The robot, the task and cup-and-ball objects as well as the learning algorithms remained the same as in Study 1. Instead of the interfaces for human feedback, a camera setup was used to determine the costs by measuring the minimum distance between the ball and the cup.

1) Camera Setup: To calculate the minimum distance from ball to cup for each sample, a setup with two cameras was put up using two Logitech webcams with 60 fps (see Fig. 6). The setup was similar to the one of a previous study [3]. One camera was positioned in front of the robot and oriented towards the ball-and-cup (front camera). A second camera was positioned above the robot, again oriented towards the ball-and-cup (top camera). To synchronize both cameras, a magenta screen was shown on the robot's tablet before the sample execution. For calculating the minimal distance between the ball and cup, first, the front camera detected the ball and the cup for each frame. At the exact frame, where the descending ball crosses the top rim of the cup, the euclidean distance between the center of the cup and the center of the ball from the top camera is calculated.

2) Optimization: In Study 2, for both optimization conditions, the same initial demonstration was used and all hyperparameters remained as shown in Table I. The distances between ball and cup were used as costs for each roll-out in a batch for the *absolute scale* optimization approach. In the *preference-based approach*, the two respective distances were compared and the movement with the shorter distance propagated.

B. Study Design

For each of the two optimization approaches, *absolute scale* and *preference-based*, we recorded 30 optimization runs with the same abort criteria (five consecutive hits or a maximum of 80 roll-outs) as in Study 1. The quantitative measures were calculated as in Study 1: Interaction Time, First Hit, and Hit Ratio.

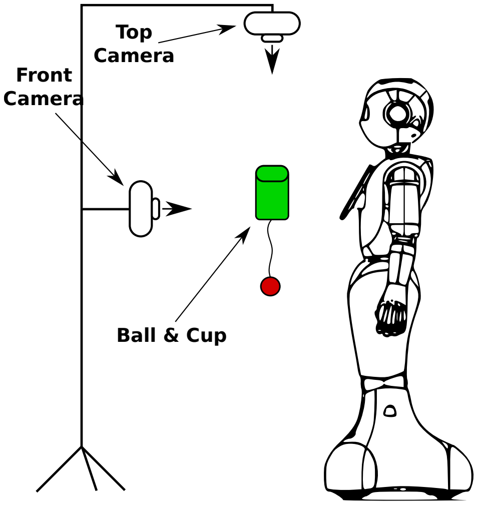
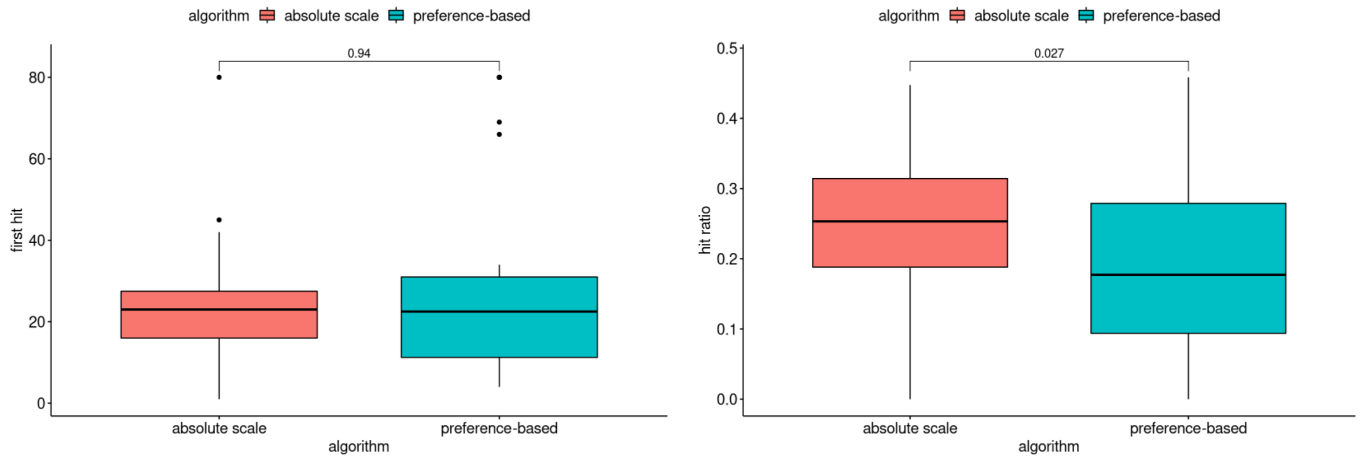


Fig. 6: Camera setup.

C. Results

The quantitative results of *first hit* and *hit ratio* are depicted in Figure 7. The system with the *preference-based* feedback resulted in no significantly earlier first hit (M = 27.9, SD = 23.46) compared to the *absolute scale* (M = 24.07, SD = 14.74) feedback system, W = 444.5, p = .94. However, the overall ratio of hits to total roll-outs is significantly higher for the *absolute-scale* condition (M = .25, SD = .1) compared to the *preference-based* condition (M = .19, SD = .12), W = 600, $p < .05$.

The total interaction time between the conditions was not significantly different, W = 407, p = .53. In the *preference-based* condition, each optimization run took on average M = 1525.8 seconds (SD = 607.81) and in the *absolute scale*



(a) Number of roll-outs needed for the first hit.

(b) Number of hits to total number of roll-outs ratio.

Fig. 7: Results for Study 2: algorithm comparison.

condition, each run took M = 1395.03 (SD = 355.44) seconds on average.

D. Discussion

The results for the measure *hit ratio* reveal that with objective, camera-based costs the *preference-based* optimization does not perform as well as the *absolute scale* optimization. For the remaining measures, differences were not significant, however, means for these measures also point in the same direction. The differences found in Study 1 can thus be attributed to the human in the loop. Not only were human users able to equally use both optimization approaches but they were much more proficient in providing *preference-based* feedback. The differences in algorithmic performance in Study 1 with user ratings and Study 2 with objective camera setup reveal that interactive task learning demands new perspectives on algorithm design: Besides the objective performance of an algorithm, factors of comprehensibility and transparency for users need to be considered. Through a more intuitive understanding of the functionality of an algorithm, users will be more proficient in teaching a robot a new skill.

A potentially critical missing research direction for this research is the investigation of long-term effects in HRI teaching scenarios. Differences in the effectiveness and usability of the feedback types might only be revealed by long-term studies in which participants have to teach the robot diverse tasks over an extended time. We hypothesize that in long-term teaching scenarios, the *preference-based* feedback will be easier to use for the user and less annoying than the *absolute scale* feedback.

IV. Conclusion

In conclusion, we have investigated the subjective and objective difference of using either *absolute scale* or *preference-based* user feedback for an interface to teach a robot a new skill of the game cup-and-ball. We conducted a between-subjects experiment to evaluate the differences between the interfaces in terms of the user's subjective evaluation and on the actual algorithm performance and a follow-up study to compare the two algorithms with objective costs obtained via a camera setup. Our results show that the users in our experimental setup do not indicate a preference for one or the other kind of feedback. The participants in both conditions equally evaluate the robot and the task. However, the system using *preference-based* optimization performed better in terms of learning the task faster and having more successful roll-outs with human feedback, despite falling behind the *absolute scale* optimization in terms of the number of successful roll-outs with objective costs instead of human feedback. These performance differences emphasize the importance of taking into account the human and factors of comprehensibility and transparency when developing and evaluating learning algorithms for interactive task learning.

This present work is, to the best of our knowledge, the first work investigating *preference-based* feedback to teach a physical robot a new skill with non-expert users. The fact that teaching through preference-based feedback was more successful and faster presents evidence that this approach to teach a robot a new skill is useful and more intuitive than other approaches.

Appendix A
Details of the Radial Basis Function network implementation

We approximate the sequence $(\mathbf{x}_t)_{t=1}^{T}$ by means of a Radial Basis Function Network (RBFN) that takes as input a scalar representation of time, $t \in [0, T]$, s.t.

$$\mathbf{x}_t = \mathbf{x}_0 + f(t, W) = \mathbf{x}_0 + W \cdot \mathbf{h}(t). \tag{3}$$

where $\mathbf{h}(t) = (h_1(t), \ldots, h_n(t))$ is a vector of n fixed radial basis functions evaluated at t, and $\mathbf{x}_0$ is the arm configuration at the beginning of the movement. As basis functions, we use a set of Gaussian functions,

$$h_i(t) = \exp\left(-\frac{(t - c_i)^2}{2a^2}\right), \tag{4}$$

where c_i is the center of the Gaussian function, and the scalar constant a controls its width. We choose the centers c_i in such a way, that they are evenly spread across the interval $[\delta, T-\delta]$, where the offset δ is used to ensure that the radial basis functions evaluate close to zero at the start and the end of the movement ($t = 0$ and $t = T$, respectively). This way, we can easily ensure that no unstable behavior of the robot occurs, as the arm begins in, and returns to, the configuration $\mathbf{x}_0$, since $f(t, W) \approx \mathbf{0}$ for $t = 0$ and $t = T$. Specifically, we set $\delta = \frac{10 \cdot T}{27}$.

To generate a smooth trajectory, the basis functions must be sufficiently overlapping. We empirically selected $a = \frac{T}{9}$ to obtain a suitable width for the Gaussian basis functions.

References

[1] A. Bharadwaj, M. Dvorkin *et al.*, "The rise of automation: How robots may impact the us labor market," *The Regional Economist*, vol. 27, no. 2, 2019.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Survey: Robot programming by demonstration," *Handbook of robotics*, vol. 59, no. BOOK_CHAP, 2008.

[3] A.-L. Vollmer and N. J. Hemion, "A user study on robot skill learning without a cost function: Optimization of dynamic movement primitives via naive user feedback," *Frontiers in Robotics and AI*, vol. 5, p. 77, 2018.

[4] E. Brochu, T. Brochu, and N. de Freitas, "A bayesian interactive optimization approach to procedural animation design," in *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 2010, pp. 103–112.

[5] A. Cockburn and C. Gutwin, "Anchoring effects and troublesome asymmetric transfer in subjective ratings," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[6] K. Zupanc and E. Štrumbelj, "A bayesian hierarchical latent trait model for estimating rater bias and reliability in large-scale performance assessment," *Plos one*, vol. 13, no. 4, p. e0195297, 2018.

[7] D. C. Kingsley, "Preference uncertainty, preference refinement and paired comparison choice experiments," *Dept. of Economics. University of Colorado*, 2006.

[8] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[9] A. L. Thomaz, C. Breazeal *et al.*, "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance," in *Aaai*, vol. 6. Boston, MA, 2006, pp. 1000–1005.

[10] M. Cakmak and A. L. Thomaz, "Optimality of human teachers for robot learners," in *2010 IEEE 9th International Conference on Development and Learning*. IEEE, 2010, pp. 64–69.

[11] A. Najar and M. Chetouani, "Reinforcement learning with human advice. a survey," *arXiv preprint arXiv:2005.11016*, 2020.

[12] E. Senft, S. Lemaignan, P. E. Baxter, and T. Belpaeme, "Leveraging human inputs in interactive machine learning for human robot interaction," in *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2017, pp. 281–282.

[13] W. B. Knox, B. D. Glass, B. C. Love, W. T. Maddox, and P. Stone, "How humans teach agents," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 409–421, 2012.

[14] W. B. Knox and P. Stone, "Reinforcement learning from human reward: Discounting in episodic tasks," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012, pp. 878–885.

[15] C. Wirth, R. Akrour, G. Neumann, and J. Fürnkranz, "A survey of preference-based reinforcement learning methods," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 4945–4990, 2017.

[16] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions." in *Robotics: Science and Systems*, 2017.

[17] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, 2017, pp. 4299–4307.

[18] A. Weiss, J. Igelsbock, S. Calinon, A. Billard, and M. Tscheligi, "Teaching a humanoid: A user study on learning by demonstration with hoap-3," in *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2009, pp. 147–152.

[19] J. Kober and J. Peters, "Learning motor primitives for robotics," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2112–2118.

[20] F. Stulp, "`DmpBbo` – a c++ library for black-box optimization of dynamical movement primitives." 2014. [Online]. Available: https://github.com/stulp/dmpbbo.git

[21] J. Matyas, "Random optimization," *Automation and Remote control*, vol. 26, no. 2, pp. 246–253, 1965.

[22] C. M. Carpinella, A. B. Wyman, M. A. Perez, and S. J. Stroessner, "The robotic social attributes scale (rosas): Development and validation," in *Proceedings of the 2017 ACM/IEEE International Conference on human-robot interaction*. ACM, 2017, pp. 254–262.

[23] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[24] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*. Elsevier, 1988, vol. 52, pp. 139–183.

[25] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org/

[26] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.