Robot Skill Learning with User Feedback: evaluating system performance and human factors*

Anna-Lisa Vollmer[1] and Nikolas J. Hemion[2]

***Abstract*— Enabling users to teach their robots new tasks at home is a major challenge for research in personal robotics. This work presents a user study in which participants were asked to teach the robot Pepper a game of skill. The robot was equipped with a state-of-the-art skill learning method, based on dynamic movement primitives. The only feedback participants could give was a discrete rating after each of Pepper's movement executions ("very good", "good", "average", "not so good", "not good at all"). We compare the learning performance of the robot when using user-provided feedback with a version of the learning where an objectively determined cost function is used. Our results show that a) it is possible to optimize a complex skill with such simple discrete feedback, and b) un-experienced users with no knowledge about the learning algorithm naturally tend to apply a working rating strategy, leading to similar learning performance as when using the objectively determined cost. We provide insights about difficulties when learning from user provided feedback, and make suggestions how the learning could be improved.**

I. INTRODUCTION

Robots are currently making their entrance in our everyday lives. To be able to teach them everyday tasks, learning mechanisms need to be intuitively usable by everyone. The general aim of this work is to understand if state-of-the-art learning algorithms are compatible with human non-expert users' teaching behavior. The field of Interactive Machine Learning (IML) aims to give the human an active role in the machine learning process. It is a rather vast field including the human in an interactive loop with the machine learner: the learner shows its output (e.g. performance, predictions) and the human provides input (e.g. feedback, corrections, examples, demonstrations, ratings). In robotics, it combines research on machine learning and human-robot interaction. Though there exists related work in other areas, for example on concept learning [1], [2], this work particularly concentrates on policy learning with a human teacher: allowing the user to teach their robot a new task.

IML in robotics has mainly been applied to virtual agents. Work that has been done with physical robots has so far only investigated scenarios involving learning of discrete sets of actions. Furthermore, in typical HRI studies with complex state-of-the-art learning systems, evaluation with the human in the loop is often done by the system developers themselves.

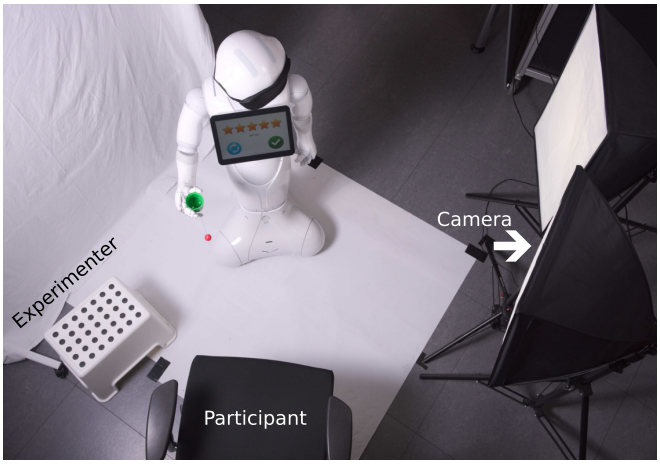


Fig. 1: Experimental setup from above. In the studies with optimization via the external camera setup (Section II-B), where the experimenter only returned the ball to its home position, the seat for the participant remained empty.

Related work in this area includes the work of Thomaz and colleagues, who investigated user input to a reinforcement learning agent that learns a sequential task in a virtual environment [3]. They then altered the learning mechanism according to the results of their Human-Robot Interaction (HRI) studies. Also Senft et al. recently presented a study with a virtual reinforcement learning agent learning sequential tasks with user rewards [4]. Knox and colleagues developed a framework where an agent's policy is shaped by human reinforcement signals, the 'Training an Agent Manually via Evaluative Reinforcement' (TAMER) framework [5]. TAMER is based on Q-learning and builds a model of the human reward. It is mostly used for learning of discrete tasks, but has also been applied to simulations of rather simple continuous tasks [6]. In TAMER the learner is additionally given environmental rewards. Very recently, Christiano et al. have also focused on teaching reinforcement learning agents novel behaviors [7]. They successively presented pairs of short video clips showing the performance of virtual agents (simulated robots in one task, and agents playing Atari games in another task) to human participants, who then selected the performance that they preferred. Using this feedback alone, the virtual agents were able to learn interesting behaviors, without relying on environmental rewards, as opposed to TAMER. However, their work is based on

*A.-L. V. was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

[1]Anna-Lisa Vollmer is with Faculty of Technology, Cognitive Interaction Technology Center of Excellence, Bielefeld University, Germany avollmer@techfak.uni-bielefeld.de

[2]Nikolas J. Hemion is with SoftBank Robotics Europe, Paris, France nhemion@softbankrobotics.com

deep reinforcement learning methodology and thus requires the agent to train for hundreds of hours, which poses a severe difficulty for application in real robots, on the one hand in terms of time necessary for training, and on the other hand due to other factors such as physical wear down. Interestingly, they were able to reduce the amount of human feedback necessary to only about one hour. In contrast to Christiano et al., here, we perform experiments with a real physical robot and successful movements can be learned in about 30 minutes.

In this work, we present a first study with naive non-expert participants who teach a full-size humanoid robot equipped with a completely unadjusted state-of-the-art learning mechanism a complex movement skill. Importantly, the movement involves continuous motor commands and cannot be solved using a discrete set of actions.

We use Dynamic Movement Primitives (DMPs), which are "the most widely used time-dependent policy representation in robotics [8], [9]." ([10], p.9), combined with Covariance Matrix Adaptation Evolution Strategy (CMA-ES) for optimization. The task to be learned is the ball-in-cup game as described by Kober et al. [11]. Usually, these state-of-the-art learning mechanisms are tested in the lab in simulation or with carefully designed cost functions and external tracking devices. Imagine robots in private households that should learn novel policies from their owners. In this case, the use of external tracking devices is not feasible, as it comes with many important requirements (e.g. completely stable setup and lighting conditions for color-based tracking with external cameras).

Absolute distances obtained via the objective function are so far the only cost with which policy search has been successfully tested. However, it is difficult for humans to give absolute distances (i.e., the cost) as feedback to the robot. Therefore, we let participants in our study give discrete feedback on a scale from one to five.

The central question we aim to answer is: Can we use simple CMA-ES based algorithms to train movement skills with unexperienced users providing discrete feedback?

We will also identify important difficulties of making learning in this setup work with an external camera setup and with human users.

For the evaluation, we focus on system performance and the user's teaching behavior.

II. METHOD

A. System

1) Robot: Pepper is a 1.2 m tall humanoid robot developed and sold by SoftBank Robotics. Pepper's design is intended to make the interaction with human beings as natural and intuitive as possible. It is equipped with a tablet as input device. Pepper is running NAOqi OS. Pepper is currently welcoming, informing and amusing customers in more than 140 SoftBank Mobile stores in Japan and it is the first humanoid robot that can now be found in Japanese homes.

Pepper used only its right arm to perform the movements. The left arm and the body were not moving. For the described studies, any collision avoidance of the robot has been disabled. Joint stiffness is set to 70%.

2) Setup: The setup is shown in Fig. 1. Two cameras recorded the movement at 30 Hz, one from above and another one from the side. This allowed for tracking of the ball and cup during the movements. All events, including touch events on the tablet of the robot were logged.

3) Ball and cup: The bilboquet (or ball and cup) game is a traditional children's toy, consisting of a cup and a ball, which is attached to the cup with a string, and which the player tries to catch with the cup. Kober et al. have demonstrated that the bilboquet movement can be learned by a robot arm using DMP-based optimization [11], and we have demonstrated that Pepper is capable of mastering the game[1]. In this study, the bilboquet toy was chosen such that the size of the cup and ball resulted in a level of difficulty suitable for our purposes (in terms of time needed to achieve a successful optimization) and feasibility regarding the trade-off between accuracy (i.e. stiffness value) and mitigating hardware failure (i.e. overheating). Usually, such a movement optimization provides a more positive user experience when learning progress can be recognized. Thus, the initialization and exploration parameters together should yield an optimization from movements somewhere rather far from the cup toward movements near the cup. With a small cup, if the optimization moves rather quickly to positions near the cup, the 'fine-tuning' of the movement to robustly land the ball in the cup takes disproportionally long. This is partially due to the variance introduced by hardware. Therefore, we chose the cup size to result in a agreeable user experience by minimizing the time spent on "fine tuning" of the movement near the cup at the end of the optimization process on the one hand, and on the other hand by minimizing the teaching time until the skill has been successfully learned.

4) Learning algorithm: For learning the ball-in-a-cup skill on Pepper, we adopt Stulp and Sigaud's method [12] of optimizing a dynamic movement primitive representation [13] using simple black-box optimization [14]. More specifically, we use Covariance Matrix Adaptation Evolution Strategy (CMA-ES) for optimization. The parameter space is 150 dimensional as we use 5 degrees-of-freedom (DoF) in the robot arm and 30 DMP parameters. The DMP is parameterized by a locally weighted regression to represent the forcing term [12] using 30 local models for each DoF. For each local model, we only optimize a single parameter, which is the offset of the local model. CMA-ES functions similarly to a gradient descent. After the cost has been obtained via the defined objective function for each roll-out in a batch, in each update step, a new mean value for the distribution is computed by ranking the samples according to their cost and using reward-weighted averaging. New roll-outs are sampled according to a multivariate normal distribution in $\mathbb{R}^n$ with

[1] https://youtu.be/jkaRO8J_1XI

(a) (b)

Fig. 2: Detection of ball and cup at the respective frame of interest in side and top view.

here, $n = 150$. There are several open parameters which we manually optimized. We aimed at allowing a convergence to a successful movement within a reasonable amount of time. The parameters include the initial trajectory given to the system as a starting point, the number of basis functions the DMP uses to represent the movement, the initial covariance for exploration and the decay factor by which the covariance is multiplied after each update, the batch size as the number of samples (i.e. roll-outs) before each update, the stiffness of the joints of the robot, the number of batches (i.e. updates) for one session in the described studies. The initial trajectory was recorded via kinesthetic teaching to the robot. We chose a trajectory with too much momentum, such that the ball traveled over the cup. 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
Initialization	Same for all studies.
Number of basis functions	30
Covariance	80
Decay rate	0.8
Batch size	10
Stiffness	70 %
Number of batches	8

B. OPTIMIZATION – EXTERNAL CAMERA SETUP

In order to optimize the movement with external cameras, a cost function is defined that determines the cost as the distance between the ball and the cup at height of the cup when the ball is traveling downward, similar as described in [11]. During a roll-out, the ball typically (this depends on the chosen initialization, here, it will) passes the height of the cup and then descends again. From a webcam recording the side of the movement, we determine the exact frame when the descending ball passes the vertical position of the cup. In the corresponding frame from the top view camera at this moment, we measure the distance between the center of the ball and the center of the cup in pixels (see Fig. 2).

We showed a cyan screen right before the movement began which could be detected automatically to segment the video streams. The experimenter repositioned the ball in the home position after each roll-out.

Apart from the usual issues for color-based tracking, as for instance overall lighting conditions, the above simple heuristic for cost determination needed several additional rules to cover exceptions (for instance, dealing with the ball being occluded in the side view when it lands in the cup or passes behind the robot's arm). More severely, in this particular task the ball occasionally hits the rim of the cup and bounces off. The camera setup in this case detects the frame in which the ball passes beside the cup *after* having bounced off the rim, and thus assigns a too high cost to the movement. Although we were aware of this, we restrained from taking further measures to also cover this particularity of the task, as we found that the camera-based optimization would still succeed. In a version of the game with a smaller cup size however, this proves to be more problematic for the optimization and needs to be taken into account.

For initial trajectories that do not reach the height of the cup, additional rules would need to be implemented for low momentum roll-outs.

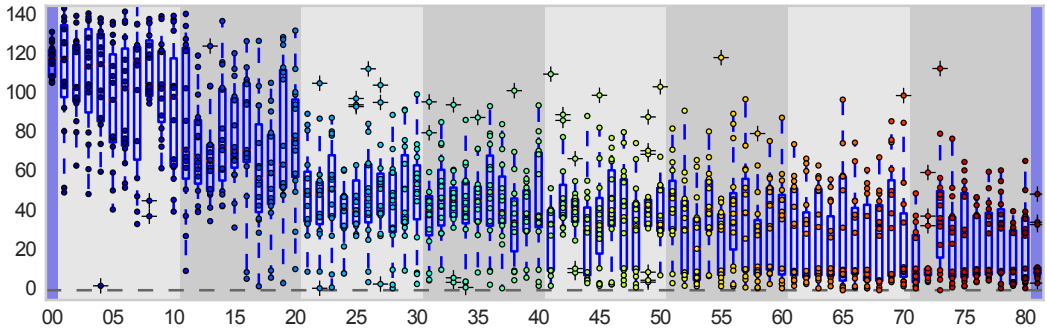
C. OPTIMIZATION – NAIVE USERS

In the following, we describe the conducted HRI study with non-expert users, who are naive to the learning algorithm and have little to no experience with robots. It was approved by the local ethics committee.

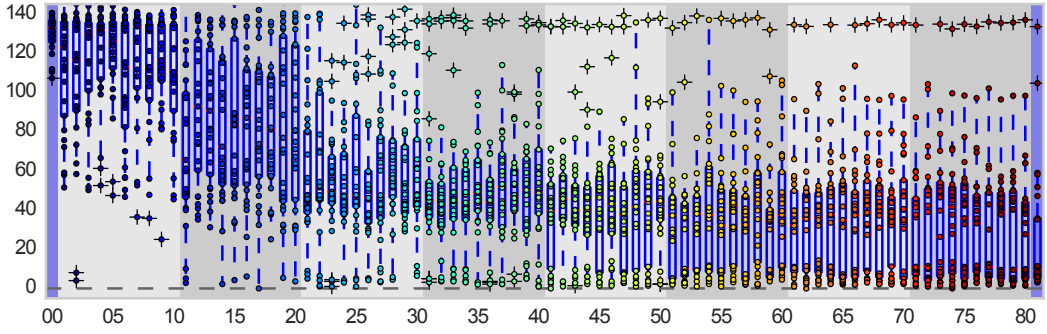
1) Participants: Participants were recruited through flyers/adds around the campus of Bielefeld University, at children's daycare centers, and gyms. Twenty-six persons took part in the experiment. Participants were age- and gender-balanced (14 f, 12 m, age: $M = 39.32$, $SD = 15.14$ with a range from 19 - 70 years).

2) Experimental Setup: 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.1). As in the other two conditions, two cameras recorded the movement, one from above and another one from the side, such that a ground truth cost could be determined. However, the camera input was neither used for learning, nor was it communicated to participants how the cost would be determined from the camera images. Informed consent was obtained from all participants prior to the experiment.

3) Course of the experiment: Each participant was first instructed (in German) by the experimenter. The instructions constitute a very important part of the described experiment because everything that is communicated to participants about the robot and how it learns might influence the participants' expectations and, in turn, their actions (i.e. ratings). Therefore, the instructions are described in full detail. It included the following information: The research conducted is about robot learning. The current study tests the learning of the robot Pepper and if humans are able to teach it a task, especially a game of skill called ball in cup. The goal



(a) Ground truth for camera optimized sessions.



(b) Ground truth for naive user optimized sessions.

Fig. 3: Ground truth from cameras for the 80 roll-outs in a session. First and last movements (with blue background) are initialization and final mean, respectively. Gray backgrounds indicate batches (8 in total). The central mark of box plots is the median, the lower edge of a box is the 25th percentile and the upper edge the 75th percentile, the whiskers extend to 1.5 times the interquartile range. Dots with underlying crosses lye outside the whiskers and could be considered outliers. Successful movement executions can clearly be distinguished from unsuccessful ones, as they lie in a "band" of distance costs between 0 and around 15, corresponding to the ball lying inside the cup. The ball passing directly next to the cup resulted in a computed cost larger than 20, resulting in the clear separation that can be seen.

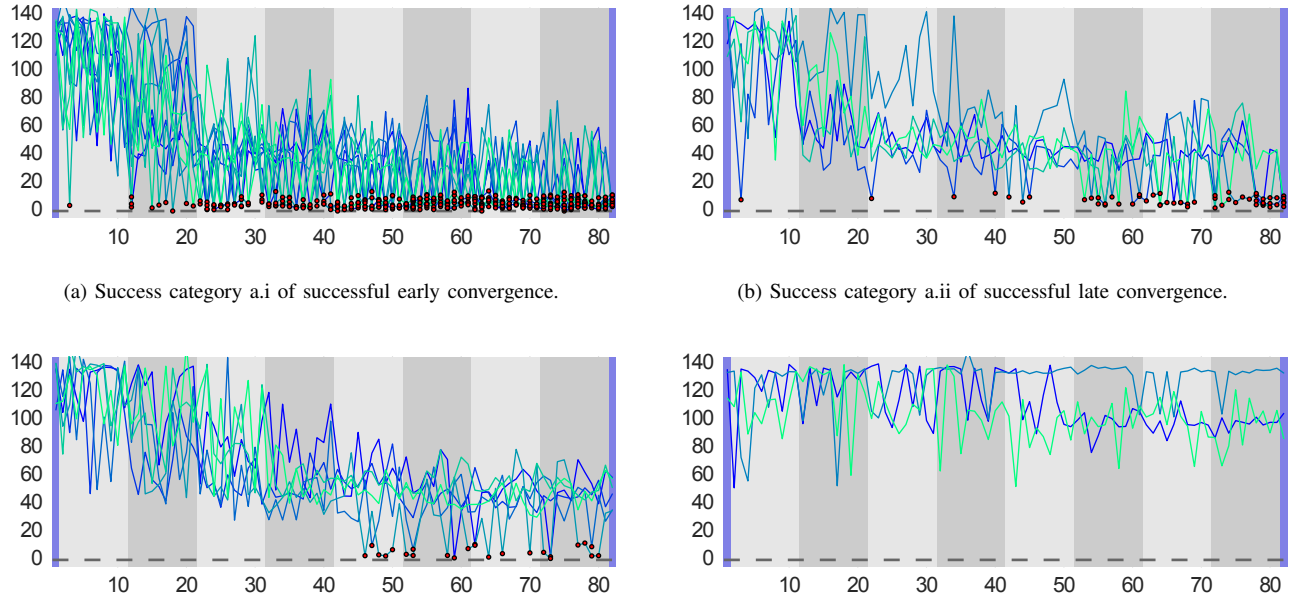
of the game is that Pepper gets the ball into the cup with movement. During the task, Pepper will be blindfolded. The cup is in Pepper's hand and in the home position the ball is hanging still from the cup. The participant was instructed that he/she could rate each movement via a rating GUI, which was displayed on the robot's tablet. The experimenter showed and explained the GUI. The participant can enter up to 5 stars for a given roll-out (as in 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. A rating is confirmed via the green check mark button on the right. Another button, the replay button on the left, permitted the participant to see a movement again, if needed. When the rating was confirmed, it was transformed into a cost as $\text{cost} = 6 - \text{rating}$ to invert the scale, and was associated to the last shown movement for the CMA-ES minimization. A ready prompt screen was then shown to allow the repositioning of the ball still in the home position. After another button touch of confirmation on this screen, the robot directly showed the next roll-out.

The camera-setup remained the same also in this study, however, the videos were only saved and used afterwards to compute ground truth. In this study, the cameras were not part of cost computation or learning. Participants were also informed of the cameras recording the movements and that each participant does a fixed number of ratings at the end of which the tablet will show that the study has ended. At this point, participants were encouraged to ask any potential questions they had.

Neither did we tell participants any internals of the learning algorithm, nor did we mention any rating scheme. We also did not perform any movement to prevent priming them about correct task performance.

Then, Pepper introduced itself with its autonomous life behavior (gestures during speech and using face detection to follow the participant with its gaze). Pepper said that it wanted to learn the game blindfoldedly but did not know yet how exactly it went. It further explained that in the following it would try multiple times and the participant had to help it by telling it how good each try was. After the experimenter had blindfolded Pepper, the robot showed the movement of the initialization (see Section II-A.4).

After rating the 82 trials (the initialization + 80 generated roll-outs + the final optimized movement), each participant



(a) Success category a.i of successful early convergence.

(b) Success category a.ii of successful late convergence.

(c) Success category b of premature convergence.

(d) Success category c of unsuccessful convergence.

Fig. 4: System performance for all sessions in a success category. Each line corresponds to camera obtained ground truth (i.e., automatically detected ball to cup distance) for one session. Dots mark hits.

filled out a questionnaire on the usability of the system, and the participant's experience when teaching Pepper. A short interview was conducted that targeted participants' teaching strategies and feedback meaning.

III. Experimental Results

A. System Performance

The system performance in the two studies is shown in Fig. 3. To compare the system performance across the studies, we defined five different measures of success on the objective cost only:

- Is the final mean a hit or a miss? (Final.hit)
- The distance of the final mean in pixels (Final.dist)
- The mean distance of all roll-outs in the final batch in pixels (Batch.dist)
- The total number of hits (#hits)
- The number of roll-outs until the first hit (First.hit)

Based on these success measures, we perform statistical tests with the aim to determine what is more successful in optimizing this task, the camera setup or the naive users.

The tests did not reveal any significant differences in performance between the two. Descriptive statistics can be found in Table II. We conducted a CHI-square test for the binary hit or miss variable of the final roll-out (Final.hit) which did not yield significant results, $\chi^2(1, 41) = 1.5, p = 0.221$. We conducted four independent samples t-tests for the rest of the measures. For the distance of the final mean (Final.dist), results are not significant, $t(35.66) = -1.527, p = 0.136$. For the mean distance in roll-outs of the final batch (Batch.dist), results are not significant, $t(39) = -0.594, p = 0.556$. For the total number of hits (#hits), results are not significant, $t(39) = 0.66, p = 0.513$. For the number of roll-outs until the first hit (First.hit), the analysis was not significant either, $t(31) = -0.212, p = 0.834$.

TABLE II: Descriptive Statistics

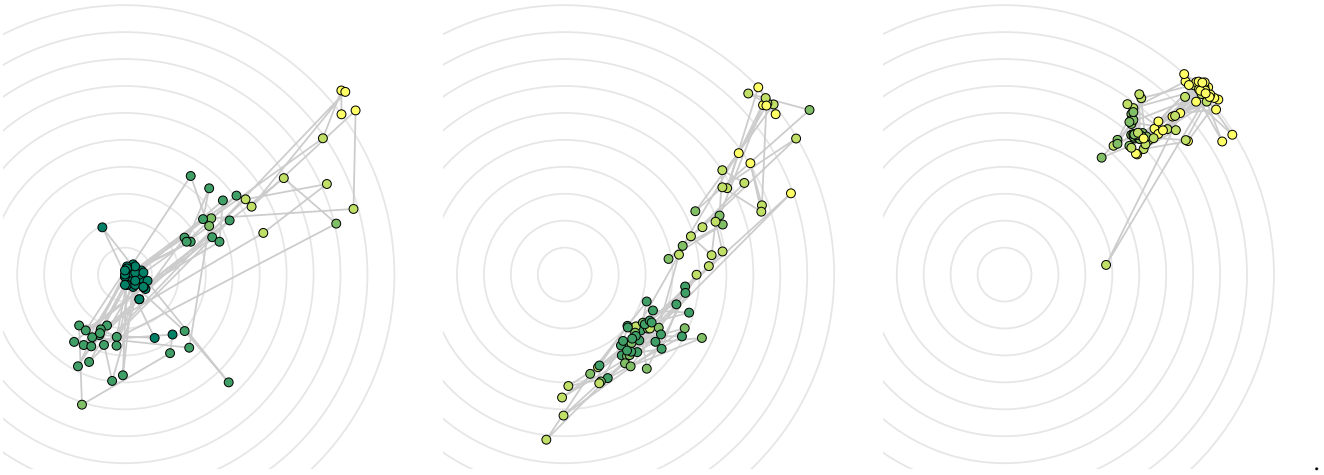
Measure	Cam		HRI	
Final.hit	80%	hits	61.5%	hits
	$M =$	$SD =$	$M =$	$SD =$
Final.dist	14.39	11.21	21.89	20.15
Batch.dist	25.88	16.00	27.82	21.66
#hits	20.27	11.84	17.96	14.97
First.hit	27.15	17.01	28.55	19.41

When looking at the HRI study only, we identify three main cases of learning performance: a) successful convergence, with sub-cases a.i) early convergence, $N = 12$ and a.ii) late convergence, $N = 5$; b) premature convergence, $N = 6$; and c) unsuccessful convergence, $N = 3$ (see Fig. 4). Also in the camera optimized sessions, two out of 15 sessions showed unsuccessful convergence, which hints at important difficulties in both setups.

B. User Teaching Behavior

To investigate the teaching behavior of the non-expert users, we are particularly interested in the strategies that are successful or unsuccessful for learning.

1) Questionnaire and Interview: We first report the questionnaire and interview answers relating to the strategies of the participants in our study. This will give us a general idea about their (self-reported) teaching behavior before we



(a) (b) (c)

Fig. 5: Individual visualizations for all roll-outs in one prototypical session for (a) successful, (b) premature, and (c) unsuccessful convergence. Colors show score given (darker shades correspond to higher scores, brighter shades correspond to lower scores). Concentric circles show equidistant positions around the cup, which is located in the center.

analyze the actual scores. The strategies participants report in questionnaires and interviews can be categorized into five approaches.

a) Distance from ball to cup: The majority of participants (N = 15) reported to use scores to rate the distance from the ball to the cup. Interestingly, all of these participants are part of sessions we identified as (a) successful convergence. This suggests that this strategy leads to success.

b) Momentum: A few participants (N = 2) reported to rate the momentum of a movement. Of course at the beginning of the sessions, the momentum correlates with the distance of the ball and cup. A movement with less momentum moves the ball closer to the cup. One of the participants who reported this strategy successfully trained the robot, for the other participant, the exploration converged prematurely.

c) Comparative ratings: A few others (N = 4) reported to give ratings comparing each movement to the previous one: if the movement was better then before, the rating was better and vice versa. Interestingly, sessions of participants with this teaching strategy all fall into the premature convergence category (b) described in Subsection III-A.

d) Spontaneous ratings: Two participants claimed to rate the movements spontaneously, without any clear strategy (N = 2). For one of the two participants, exploration converged late, but successfully (a) and for the other the session was unsuccessful (c).

e) Individual strategies: The remaining participants reported individual strategies (N = 3). For instance one participant in this category gave always the same score (one star) with the intention to let the robot know that it should try something completely different in order to change the movement completely. The other two strategies were not reported clearly. However, the described strategy as well as another in this category, were not successful (c). One of the participants used a strategy that lead to premature convergence (b).

2) Correlation with Ground Truth: Based on the self-reported user strategies, we expect the successful sessions to also reflect the 'Distance from ball to cup' strategy in the actual scores participants gave. We test this by calculating the correlation between the participant scores and the ground truth of the robot movements. In the HRI case in general, participants received an average correlation coefficient of $M = 0.72, SD = 0.20$. The strategy to rate according to the distance between the ball and the cup should yield a high correlation value and thus we expect successful sessions to obtain a higher correlation coefficient than sessions with premature convergence, which in turn receives a higher correlation coefficient than unsuccessful convergence (i.e., success category $a > b > c$). Because of small sample sizes, we conduct a Kruskal-Wallis H test. There was a statistically significant difference in correlation coefficients between the three different success categories, $\chi^2(2) = 8.751, p = 0.013 < 0.05$. An inspection of the mean ranks for the groups suggest that the successful sessions (a) had the highest correlation ($mean\ rank = 16.24, M = 0.75, SD = 0.20$), with the unsuccessful group (c) the lowest ($mean\ rank = 2.67, M = 0.58, SD = 0.29$), and prematurely converged sessions in between ($mean\ rank = 11.17, M = 0.045, SD = 0.25$). Pairwise post hoc comparisons show a significant difference between the successful (a) and unsuccessful (c) sessions only ($p = 0.014 < 0.05$, significance value adjusted by Bonferroni correction for multiple tests). Thus the results confirm our hypothesis.

3) Score data: Prototypical plots for the three success strategies are shown in Fig. 5. They corroborate and illustrate the teaching strategies we found.

Looking at individual plots of scores, we can draw a number of additional qualitative observations:

a) Hits receive always 5 stars.: We observe that a hit (i.e., the ball lands in the cup) for all participants always receives a rating of 5 stars. Though some participants reserve the 5 star rating for hits only, in general, also misses could receive a rating of 5.

b) Rating on a global scale: One strategy we observe is to give ratings on a global scale, resulting in scores similar to the ground truth, but discrete.

c) Rating on a local scale: Some people that rate according to the distance between ball and cup, take advantage of the full range of possible scores during the whole session and adjust their ratings according to the performance.

d) Giving the same score multiple times: Some participants gave the same score multiple times in one batch. This could be due to perceptual difficulties. Participants often complained during the study that all movements look the same. Also this behavior could be part of a specific strategy, for example a behavior emphasizing the incorrect nature of the current kind of movement in order to get the robot to change the behavior completely (increase exploration magnitude) or a strategy that focuses on something else than the distance.

IV. DISCUSSION

The contribution of this work can be summarized with two main results.

1) CMA-ES optimization with DMP representation works with un-experienced, naive users, who are giving discrete feedback.
2) The main strategy users naturally apply, namely to rate according to the distance between the ball and the cup, is most successful. Relational feedback users provide, which depicts a binary relation of preference in a pair of consecutive trials, in this setup leads to premature convergence.

The discrete feedback users provide, seems to work as well as the camera setup. We believe that with a few instructions to users, system performance in this case can even be improved and failed sessions can be prevented. We could imagine the naive users to perform even better than a cost function in some cases. For instance, hitting the edge of the cup, landing a hit by chance are easily perceived and given the correct score, whereas an algorithmic computation of the cost using a camera setup might fail to correctly track this fast and complex movement. Also if the robot performs similarly bad roll-outs for some time with the ball always at a similar distance from the cup and then for the next roll-out, the ball lands at the same distance, but on the other side of the cup, the user might give a high rating to indicate the correct direction, whereas the camera setup will measure the same distance.

The optimal teaching strategy is not known for the system in this task, but it seems that most naive users are able to successfully train the robot. However, we have observed some difficulties with the current system. The DMP representation is unintuitive for humans. During the optimization, it seems more difficult to get out of some regions of the parameter space than others. This is not apparent in the action space. Additionally, nine participants reported to have first given scores spontaneously and later developed a strategy, hinting at difficulties at the beginning of the session, because they did not have any idea how to judge the first movements as they did not know how much worse the movements could get and they did not know the magnitude of differences between movements. Apart from initial difficulties, four participants reported to be inconsistent in their ratings at the beginning or to have started out with a rating too high. This means that there is a phase of familiarization with the system.

Due to the nature of CMA-ES and the way new samples are drawn from a normal distribution in the parameter space, robot performances from one batch did not differ wildly but appeared rather similar. This was confusing to some participants, as they were expecting the robot to try out a range of different movements to achieve the task. In contrast, the CMA-ES optimization resulted in rather subtle changes to the movement. As a result, some participants rated all movements from one batch with exactly the same score. This is of course critical for the CMA-ES optimization, as it gives absolutely no information about the gradient direction.

Furthermore, with the use of CMA-ES, there is no direct impact of the ratings. Participants expected the ratings to have a direct effect on the subsequent roll-out. This lead to an exploration behavior with some participants who tested the effect of a specific rating or a specific sequence of ratings on the following roll-out. The participants reacted with surprise to the fact that after a hit, the robot again performed unsuccessful movements. The mean of the distribution in the parameter space could actually be moved directly to a hit movement, if the user had the possibility to communicate this.

The cases of premature convergence could also be prevented by instead of CMA-ES using an optimization algorithm with adaptive exploration, like $\mathrm{PI}^2_{\mathrm{CMA}}$ [15]. Furthermore, participants were in general content with the possibility to provide feedback to the robot using a discrete scale. However, several participants commented that they would have preferred to also be able to provide verbal feedback of some form ("try with more momentum", "try more to the left"). How to incorporate such feedback in the learning is subject of future work.

REFERENCES

[1] F. Khan, B. Mutlu, and X. Zhu, "How do humans teach: On curriculum learning and teaching dimension," in *Advances in Neural Information Processing Systems*, 2011, pp. 1449–1457.

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

[3] 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, 2006, pp. 1000–1005.

[4] 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.

[5] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: The tamer framework," in *Proceedings of the fifth international conference on Knowledge capture*. ACM, 2009, pp. 9–16.

[6] N. A. Vien and W. Ertel, "Reinforcement learning combined with human feedback in continuous state and action spaces," in *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–6.

[7] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *arXiv preprint arXiv:1706.03741*, 2017.

[8] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in neural information processing systems*, 2003, pp. 1547–1554.

[9] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," *Robotics Research*, pp. 561–572, 2005.

[10] M. P. Deisenroth, G. Neumann, J. Peters *et al.*, "A survey on policy search for robotics," *Foundations and Trends® in Robotics*, vol. 2, no. 1–2, pp. 1–142, 2013.

[11] J. Kober and J. Peters, "Learning motor primitives for robotics," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 2112–2118.

[12] F. Stulp and O. Sigaud, "Robot skill learning: From reinforcement learning to evolution strategies," *Paladyn, Journal of Behavioral Robotics*, vol. 4, no. 1, pp. 49–61, 2013.

[13] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

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

[15] F. Stulp and P.-Y. Oudeyer, "Adaptive exploration through covariance matrix adaptation enables developmental motor learning," *Paladyn*, vol. 3, no. 3, pp. 128–135, 2012.