Human Feedback as Action Assignment in Interactive Reinforcement Learning

SYED ALI RAZA and MARY-ANNE WILLIAMS, University of Technology Sydney, Australia

Teaching by demonstrations and teaching by assigning rewards are two popular methods of knowledge transfer in humans. However, showing the right behaviour (by demonstration) may appear more natural to a human teacher than assessing the learner's performance and assigning a reward or punishment to it. In the context of robot learning, the preference between these two approaches has not been studied extensively. In this article, we propose a method that replaces the traditional method of reward assignment with action assignment (which is similar to providing a demonstration) in interactive reinforcement learning. The main purpose of the suggested action is to compute a reward by seeing if the suggested action was followed by the self-acting agent or not. We compared action assignment with reward assignment via a user study conducted over the web using a two-dimensional maze game. The logs of interactions showed that action assignment significantly improved users' ability to teach the right behaviour. The survey results showed that both action and reward assignment seemed highly natural and usable, reward assignment required more mental effort, repeatedly assigning rewards and seeing the agent disobey commands caused frustration in users, and many users desired to control the agent's behaviour directly.

CCS Concepts: • **Theory of computation** → **Reinforcement learning**; • **Computing methodologies** → **Sequential decision making**; **Learning from demonstrations**; **Learning from critiques**; **Q-learning**; *Online learning settings*; *Markov decision processes*; • **Human-centered computing** → User studies; Web-based interaction; User interface design;

Additional Key Words and Phrases: Interactive machine learning, reinforcement learning, reward shaping, learning from human teachers

ACM Reference format:
Syed Ali Raza and Mary-Anne Williams. 2020. Human Feedback as Action Assignment in Interactive Reinforcement Learning. *ACM Trans. Auton. Adapt. Syst.* 14, 4, Article 14 (August 2020), 24 pages.
https://doi.org/10.1145/3404197

1 INTRODUCTION

Humans are accustomed to teaching a task to another human being by giving demonstrations [44]. We learn many things from others by observing them doing the task. For example, a kid may learn how to answer the doorbell just by observing an elder performing this act. In addition, Humans teach by rewarding a good act and punishing a bad act. However, human-generated rewards or punishments are not as abundant as human-provided demonstrations (our daily routine works can

The research is supported by the Australian Research Council (Grant No. DP160102693).

Authors' address: S. A. Raza and M.-A. Williams, University of Technology Sydney, Faculty of Engineering and Information Technology, 81 Broadway, Ultimo, NSW, 2007, Australia; emails: {syed.raza, mary-anne.williams}@uts.edu.au.

1556-4665/2020/08-ART14 $15.00
https://doi.org/10.1145/3404197

serve as demonstrations). It indicates that the demonstration of knowledge could be a more natural and favourable method of transferring knowledge to a learning robot than providing rewards and punishments.

In robotics, learning from demonstrations (LfD) is a popular method of robot training, where humans perform task demonstrations on idle (non-responsive) robots [5]. However, the goal of learning in a social setting should be to minimize the anti-social aspects while increasing the learning performance. Unlike LfD, humans like to have an interactive learner who responds as the learning progresses. In an interactive learning scenario, the teacher observes in-learning progress, which can help in keeping the trainer engaged and motivated. In addition, it allows the trainer to understand and improve the effectiveness of its teaching.

Reinforcement learning (RL) [47] provides a rich framework for interactive learning. Specifically, reward shaping is a method in which a trainer shapes the behaviour of a learning agent by providing instantaneous rewards to intermediate behaviours [38]. The traditional approach of reward shaping, which involves learning from human-delivered numeric rewards ("reward assignment"), has a limitation. It requires a continuous assessment of agent's actions for assigning correct rewards. This task may seem laborious and unnatural to many users. The teacher is required to be attentive to an agent's actions all the time. The trainer assigns rewards for hundreds, if not thousands, of actions. Even in a simple domain, like in Sophie's kitchen[1] [51], the training takes several minutes (the author personally trained the agent for more than 30 minutes, even then, the performance was unsatisfactory). Furthermore, the traditional method does not allow a user to demonstrate the right action to do. Instead, he/she can tell if the recent action was desirable or not.

Demonstrating the right behaviour to a learner is a natural choice in many robotic applications [5, 10]. It is similar to taking impulsive actions and does not require being attentive to the agent's actions. It is a direct mode for the demonstration of one's knowledge. Similarly, in reward shaping, it is likely that a human trainer would prefer to demonstrate what-to-do, instead of assessing agent's actions (i.e., providing reward signals). Therefore, our research question is as follows: *Does a reward shaping method in which a user assigns action labels seem more user-friendly than a method in which a user assigns reward labels?*

In this regard, we have proposed a method to learn from action labels assigned by a human trainer in a reward shaping scenario (details in Section 5). We have altered the traditional reward shaping method to take human-input as a demonstration instead of a reward signal. The resulting method of reward shaping offers to teach by acting naturally from a state. We heuristically compute the numeric reward-signal from the demonstrated behaviour by comparing the action label from the trainer with the action executed by the agent. The underlying computations for policy derivation are the same as that of the traditional method of reward shaping.

To answer the research question (as discussed above), and to compare the user-friendliness of the proposed method with the traditional method of interactive reinforcement learning, we have formulated the following four hypotheses,

(H1) We expect that users would show better task performance using action assignment than using reward assignment. The correctness of user feedback will be higher in action assignment than in reward assignment.

(H2) We expect that users will find robot training by "action assignment" more natural and easy-to-use than the training by "reward assignment." The user ratings for "action assignment" will be significantly different from that of "reward assignment."

[1] http://robotic.media.mit.edu/portfolio/sophies-kitchen/.

(H3) We expect that robot training by “reward assignment” requires more mental effort than the training by “action assignment.” Significantly more users would find “reward assignment” mentally demanding than the “action assignment.”

(H4) We expect that users would prefer “action assignment” over “reward assignment” for a future use. Significantly more users would prefer to use “action assignment” in the future.

We conducted an extensive user study on the web to evaluate these hypotheses. The users trained a simulated robot in a two-dimensional (2D) maze game and answered a post-experiment questionnaire. Besides, we examined if the user felt frustration during the training process. We found that action assignment was perceived better than reward assignment in users’ opinion according to various measures. The results of a post-training survey showed that the overall user responses were in favour of action assignment.

The next section (Section 2) provides a background to reinforcement learning and reward shaping. Section 3 provides a summary of previous researches that used reward-based methods for training an agent. Section 4 provides the details of a traditional method of reward assignment. Section 5 introduces the proposed method of reward shaping from an action assignment. Section 6 describes the test domain. Various aspects of our experimental design are discussed in Section 7. The results of the experiments are reported and analysed in Section 9. In Section 10, we have discussed and interpreted the results. Finally, in Section 11, we have provided some concluding remarks.

2 BACKGROUND

2.1 Reinforcement Learning

For a comprehensive look into reinforcement learning, we refer the reader to Reference [47]. Reinforcement learning is a learning paradigm that allows the agent to learn optimal behavior by acting in the environment. It involves two entities, an agent and an environment (as shown in Figure 1). The agent is a controller programmed to learn and execute a policy to act. The information about the current state and reward occurrence typically flows from the environment to the agent. Agent’s learning is guided by means of reward signals.

We used Q-learning, which is a popular RL algorithm, proposed by Reference [55]. It is a model-free (off-policy) method. We preferred Q-learning due to its relatively simple design. Here, we learn Q-values, $Q(s,a)$. Where, $\mathrm{s} \in \mathrm{S} = \{\mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{s}_\mathrm{N}\}$ and $\mathrm{a} \in \mathrm{A} = \{\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_\mathrm{M}\}$. The goal is to learn a policy, $\pi(s) = a$.

An update in Q-learning is given by Equation (1), where, $\alpha_t \in [0,1]$ is the learning rate, $R_t(s,a,s')$ is the instantaneous reward at time t, $\gamma \in [0,1]$ is the discount factor, s' is the next state, and a' is the next state’s action. An ϵ-greedy policy is used for action selection, which says if a random number, *rand* $\in [0,1]$, is less than $\epsilon \in [0,1]$ than a random action is taken, otherwise, an action with the highest Q-value is taken:

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha_t \left(R_t(s,a,s') + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a) \right). \tag{1}$$

2.2 Reward Shaping

Reward shaping is a method of modifying the underlying reward structure. The main objective of reward shaping is to combat the scarcity of environmental rewards. In RL, environmental rewards are generally assigned at the completion of the task or on achieving a sub-goal. Therefore, these rewards are sparse, and their effect on the intermediate action-selections may start to occur after numerous episodes of exploration. Immediate rewards are used to indicate the desirability of the recent action and to expedite learning. It shapes the policy locally such that it leads to the accomplishment of goal. Computationally, the shaping reward is added to the environmental reward,

$$\bar{R}(s,a,s') = R(s,a,s') + H(s,a,s'), \tag{2}$$

where $R(s,a,s')$ is the environmental reward, as before, and $H(s,a,s')$ is the shaping reward. Note, we denote shaping reward using H, because we used only the human-generated shaping rewards. By substituting $R(s,a,s')$ in Equation (1) with $\bar{R}(s,a,s')$ the Q-update becomes

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha_t \left(R_t(s,a,s') + H_t(s,a,s') + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a) \right). \tag{3}$$

In the previous studies, the method of providing an instantaneous reward by a human trainer was referred to as *interactive reinforcement learning* [51] or *interactive shaping* [23]. Here, a human closely observes the agent and evaluates its last action in the context of domain knowledge. Then, the trainer maps the usefulness of the recent action in the longer run to positive or negative rewards.

3 RELATED WORK

A seminal work that advocated the use of instantaneous human-generated reward signals for task-learning is the *interactive reinforcement learning* framework of Thomaz et al. [51]. Through a series of experiments with real users on a simulated domain of *Sophie's kitchen*, they identified that people want to provide feedback that can guide the agent in addition to assessing its current behavior [51]. Later, they added a guidance mechanism for future actions, which drastically accelerated learning. Suay et al. [45] further studied this framework for various state space sizes and guidance signals. A similar study compared *interactive reinforcement learning* with *behavior networks* and *confidence-based autonomy* [46]. There are other similar studies that allowed the human-trainer to directly control the reward signals to the learning agent [14, 19, 20, 26, 43].

TAMER is another principle framework to incorporate human rewards in RL [23]. TAMER learns solely from human-generated shaping rewards and can learn better policies much faster than transitional reinforcement learner in a short run. Later, to learn faster in long run, authors proposed TAMER+RL [24, 25]. Vien et al. [52] proposed an extension of TAMER, ACTAMER, to cater the continuous state-action spaces. Other studies used *socio-competitive* feedback to improve the engagement of human-trainers by integrating TAMER with an online Tetris game hosted on Facebook and by showing the agent's performance and scores in a leaderboard [33, 34]. Recently, researchers have extended TAMER for deep learning [3, 54] and to incorporate expert advice [6].

Some researchers have opted to use natural instructions to obtain rewards from a human [18, 31, 32, 49]. In Reference [31], authors showed that a 6 degree-of-freedom robot manipulator learned the pick-and-place task first using human-provided demonstrations, then improved the policy using shaping rewards provided by users as voice commands. Similarly, Tenorio et al. [49] demonstrated to learn from vocal commands, which were translated into numeric reward signals to teach a navigation task to a simulated mobile robot. In Reference [32], similar to Reference [49], they used human feedback as vocal instruction using a fixed vocabulary list to denote the desirability of learner's actions. Hwaang et al. [18] used facial expression to retrieve shaping rewards, which are used along with a designed environmental reward function.

Similar to action assignment is the notion of *Advice.* Argall et al. [4] used advice to correct parts of a trajectory of state-actions performed by the robot. In a similar approach, Goecks et al. [16] used human intervention as a control over learning agent's action-selection to avoid unsafe behaviours. Waytowich et al. [56] have introduced a cycle-of-learning framework in which training occurs first using interventions and then using rewards. Reference [29] used object-focused advice as a list of objects and associated actions, which was extracted from natural explanations about a learning task by applying sentiment analyses. Kunapuli et al. [30] used an expert human advice to correct noisy and sub-optimal demonstrations to learn using fewer trajectories in inverse reinforcement learning. Cruz et al. [11] used a parent-like trainer to provide audio-visual action advice to teach the table-wiping task to a simulated and humanoid robot.

Similar to our shaping from action assignment method (Section 5), Krening et al. [27] proposed Newtonian Action Advice (NAA) to learn from human action advice. NAA differed from our proposed method in that every time advice is given, it is followed by the agent, superseding the RL algorithm's action selection. On the contrary, we use the suggested action to compute a reward for the current state-action pair only. Moreover, NAA uses Bayesian Q-Learning, whereas our method uses a tabular Q-Learning. In a recent user study [28], the authors compared NAA with a Policy Shaping method [17]. Similar to our results, they found action advice resulted in a better user experience as compared to critique (reward). However, their measures for human experience differed from ours. Our results validate the findings of Krening et al., like, users prefer control over agent's actions and users provide almost the same amount of input in both techniques. Furthermore, our study provides three new insights, action assignment enhances users' ability to perceive the correct behaviour, repeated rewarding and agent disobeying commands are two important causes of frustration, and action assignment is less mentally demanding compared to reward assignment.

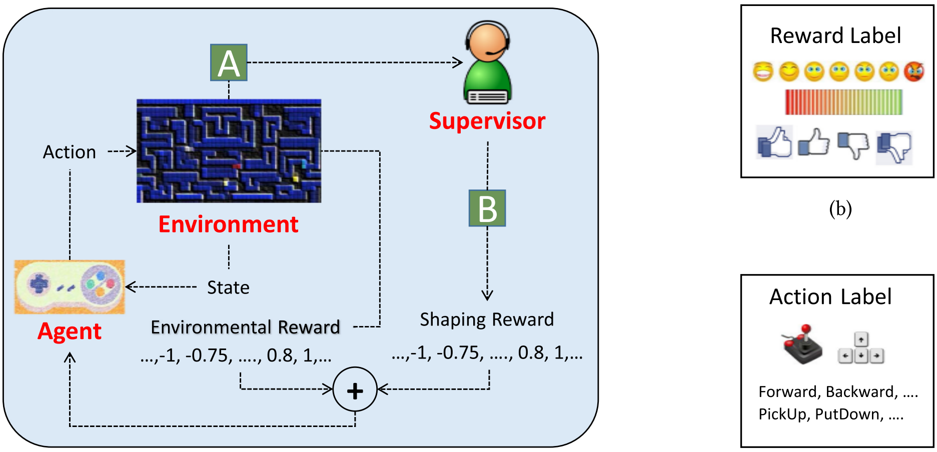
Also, our shaping from action assignment method is similar to that of Cruz et al. [12]. However, the main difference is that in their method a trainer (an artificial agent instead of a human trainer) first observes if the learning agent's current action is correct or not, and then decides if action advice should be provided or not. This puts an additional burden of assessing agent's current action at every learning step. On the contrary, the main purpose of our method is to eliminate the action assessment task.

4 SHAPING FROM REWARD ASSIGNMENT

Shaping from Reward Assignment (SfRA) is a simple method that can carry-out policy learning using arbitrary shaping rewards. It is a method that has been traditionally adopted for reward shaping (see Section 3 for related studies). The shaping rewards are assigned as signed numeric values. This method is similar to the interactive reinforcement learning method of Thomaz et al. [51]. The main difference is that SfRA does not make use of the guidance mechanism proposed in Reference [51]. Also, instead of using a continuous scale for reward assignment, we used fixed numeric rewards, a positive reward was 1, and a negative reward was -1. It helped us in keeping the experimental design simple. Note that this method will serve as a benchmark for the proposed method of shaping from action assignment, as described in the next section.

Figure 1 provides a generic view of the information flow for the two methods of Interactive RL used in this study. The reward can be taken as a label on a pre-defined scale of intensity, easily understandable to a common user (as shown in Figure 1(b)). The visual scale has an associated mapping to map the intensity labels to numeric values. The trainer observes multiple states, i.e., one before the action and one after the action as well as perceives which action was executed.

Algorithm 1 shows the basic steps of *shaping from reward assignment* method. The process starts with sampling a state as an initial state for an episode. If the agent sees a state for the first time, then it uses a random initial policy for action selection. Otherwise, an ϵ-greedy policy is used for action selection. After taking action, the agent collects the environmental reward and waits for a fixed amount of time to receive a human reward. Finally, the Q-values are updated using Equation (3), and the same process repeats in the next time step. The learning process continues until all the learning conditions are met (which was 200 steps in our case).



(a) (b) (c)

Fig. 1. (a) Information flow between different entities of the learning methods. The environment is where a physical or software-based persona of an agent takes action and can occupy a state. Also, the information about any current reward is obtained from the environment. The agent is a controller responsible to provide action commands using its policy and current state. It also updates the policy based on the (computed) rewards. The supervisor can perceive the information about the agent and the environment (box A) and provides a label (box B) from which the shaping rewards are computed. For SfRA, box A contained the information about the state before and after the action and action itself. For SfAA, box A contained information about the perceived state. Box B contained information about the labels provided by the supervisor, as shown in panels (b) and (c). (b) Examples of reward labels that can be used in SfRA. (c) Examples of action labels that can be used in SfAA.

ALGORITHM 1: Shaping from Reward Assignment

```
while learning do
    sample $s = s_{start} \in S$;
    while $s$ is not a terminal state do
        if $s$ is never visited then
            $a$ = get a random action from $A$;
        else
            $a$ = get action using $\epsilon$-greedy policy;
        end
        Execute $a$, and transition to next state $s'$, and collect environmental reward $R(s, a, s')$;
        Wait for human reward, $H(s, a, s')$;
        Update $Q(s, a)$ using Equation (3); $s = s'$;
    end
end
```

5 SHAPING FROM ACTION ASSIGNMENT

We call the proposed method of deriving shaping-reward function from action labels as *shaping from action assignment* (SfAA). The assigned actions can be seen as interactive demonstrations. It is designed after the popular interactive RL [51] but differs in how the teacher provides input to the learner. The teacher only observes the state and labels the action to choose from this state. Figure 1

ALGORITHM 2: Shaping from Action Assignment

```
while learning do
    sample $s = s_{start} \in S$;
    while s is not a terminal state do
        if s is never visited then
            $a$ = get a random action from $A$;
        else
            $a$ = get action using $\epsilon$-greedy policy;
        end
        Wait for the demonstration, $a_b$, for $X$ seconds;
        if $a_b$ provided then
            $H(s, a, s') = \Omega(a_b, a)$ from Equation (6)
        else
            $H(s, a, s') = 0$
        end
        Execute $a$, and transition to next state $s'$, and collect environmental reward $R(s, a, s')$;
        Update $Q(s, a)$ using Equation (3); $s = s'$;
    end
end
```

shows how the human-in-the-loop scenario applies to this method. Note the different types of action labels shown in Figure 1(c). Before adding to agent's policy computation, the action label is turned into a numeric value. The teaching method of action labelling (demonstration) eliminates the numeric-reward-assignment task, which is typically present in every reward shaping approach. The judgemental and evaluative task of numeric-reward-assignment may seem tedious and laborious. In addition, action assignment is more explicit as compared to rewarding in eliciting teacher's preferred policy. For the policy derivation, SfAA follows the learning mechanism of a typical reward shaping.

In an interactive learning setting, the human teacher observes agent's current state and indicates the action to select from it. The teacher's preferred policy, $\pi_t^h : S \rightarrow A$, can be written as

$$\pi_t^h(s) = a_b, \tag{4}$$

where $a_b \in A$ is defined as the best action to perform from state s as per teacher's knowledge. Note that we used t in subscript to represent that the human teacher's policy might change over time. A reward function is derived from the teacher's policy through a mapping function, $\Omega : A \times A \rightarrow R$,

$$H^d(s, a, s') = \Omega(\pi_t^h(s), \pi_t(s)). \tag{5}$$

Therefore, the problem of shaping from action assignment boils down to defining a good mapping function, Ω. The mapping function used in this work is given by the following equation:

$$\Omega(\pi_t^h(s), \pi(s)) = \begin{cases} 1, & \text{if } \pi_t^h(s) = \pi_t(s), \\ -1, & \text{otherwise.} \end{cases} \tag{6}$$

The above equation is based on a simple heuristic. A fixed positive reward is assigned if the agent's action matches with the action label by the teacher. Otherwise, a negative reward is assigned. There could be many other possibilities of defining a good mapping function.

Algorithm 2 details the complete process of SfAA. The process is similar to the one described in Algorithm 1, except that the agent waits for a fixed time-span for the trainer to provide a demonstration (action). Afterward, it computes the reward using the assigned action and takes action

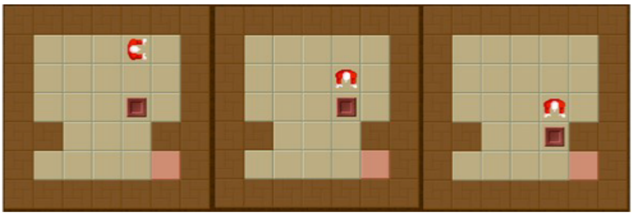


Fig. 2. The testbed domain of Sokoban. From left to right it shows the result of two consecutive down actions.

either greedily or randomly. The main trick of this method is that the action label is sought before the agent takes action. Therefore, we can compare it with the agent's estimate of correct action, and can compute the shaping reward's value. The users are required to provide feedback at each step. However, it is not mandatory. The algorithm can proceed in the absence of a user feedback at any time-step.

Note, unlike learning from demonstrations (LfD) scenario [5], in SfAA, the action selection process is under agent's control and the teacher's demonstration contributes only towards computing a reward signal for the preferred policy. Therefore, agent's initial policy may not necessarily obey the teacher's preferred policy, but later converges towards it. The method of providing action labels can be referred to as *pseudo demonstrations*, because the agent's actions are not directly controlled by the user.

Similar to this study, in our previous work [41], the online performance of action assignment was compared with a learning from critique method. The results showed that for a naïve user the policy learned via action assignment performed better than the policy learned via reward assignment. The only algorithmic difference is that in this work we use ϵ-greedy policy and in Reference [41] a greedy policy was used. Unlike this study, the users in previous study trained for a long duration (i.e., 30 episodes) and the use of greedy policy helped in increasing the convergence rate. The main difference with the previous work is in the design of the experiment. Unlike previous study, where the training was done by only one naïve user in the laboratory settings, here, the training is done by a large pool of naïve trainers recruited online. The large sample size in training has enabled drawing generalized conclusions, which was not possible in the previous study. Additionally, we have used controlled groups to test two different conditions based on the sequence of using the two training methods. Furthermore, we have used a post-training survey to capture the users' feedback on their training experiences, which was missing in the previous study.

6 TEST DOMAIN

For our experiments, we have used Sokoban as a testbed. It has an intuitive visual setting that attracts humans and keeps them fully involved while teaching. In has seen extensive use in literature for evaluation purpose [1, 2, 57]. It is a simulated 2D grid-world game where each cell can be a wall or a free cell. The free cell can be occupied by either the player or a box. The player can choose from four actions: left, right, up, and down. The player's task is to push each box using four actions (it cannot pull the box) and drive it to the goal position without letting the box get stuck, which is called deadlock. Figure 2 shows the domain and an example of gameplay. The bottom-right corner is the goal state (pinkish orange).

We have simplified the Sokoban game in terms of complexity. We modified the domain to have only one level. Instead of using multiple boxes, we choose to use only one box. The modified domain had 23 free spaces. Also, instead of using a layout from the original game, we designed the layout by ourselves. The placement of walls was simplified to avoid frequent deadlocks. Typically,

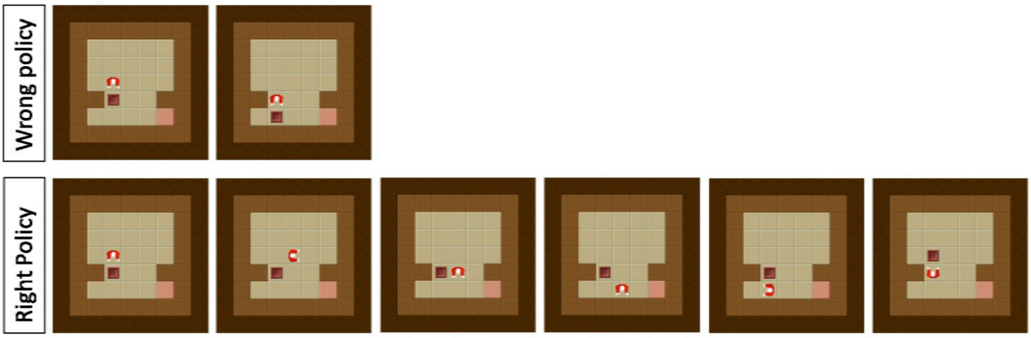


Fig. 3. The two difficult scenarios for the non-expert users. In each sub-figure, the top row shows a wrong (sub-optimal) policy and the bottom row shows an optimal policy.

at the start of the game, the player and each box are positioned at a fixed location. Instead, we have used a state distribution for both box and player for their start positions. Note that the rules of the original game were unchanged in the modified domain. A case study on the complexity of Sokoban and human problem-solving skill is provided by Reference [21]. Figure 3 shows one part of a policy that can be hard to perceive by a non-expert user. A user may think that moving the box down is right action, as it makes the box closer to the goal, but in actual, it leads to a deadlock. Since there is no pull action, the agent cannot recover. Similarly, perceiving the action(s) that can lead to the goal via shortest path (i.e., optimal action) was a hard task from many game states to a new user.

7 EXPERIMENTS

7.1 Experimental Design

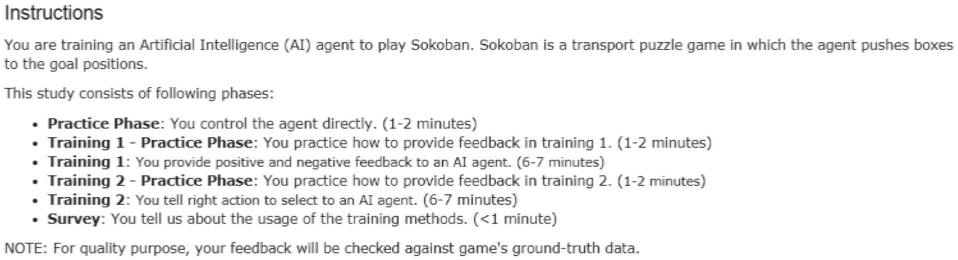
In our experiment, the independent variable was the version of the training method (SfRA or SfAA), which we manipulated. We adopted the *within-subject* experimental design where each participant was assigned to train using both the training methods. However, in such a scenario, "novelty effect" can take place [15]. If a user has never interacted with a learning robot, then the user's curiosity about robot's technology can pollute the user responses. A common practice is to have a pre-experiment session in which the user practices interactions with the robot and becomes familiar with the technology. Another important factor that can effect the quality of user responses is the familiarity with the domain. In a within-subject design, it is possible that the user gains useful insights about the domain while using the first training method. It may result in an increased rate of correct feedback when using the second training method. It is also possible that the second training method may appear easier than the first method, thus, polluting the user's judgement on the ease-of-use of the two training methods. Therefore, to counterbalance these conditions,

- half of the users trained using SfRA first and SfAA second (Condition 1)
- and the other half trained using SfAA first and SfRA second (Condition 2).

By reversing the ordering of the training methods, we ensured that these effects were observed uniformly in the data collected for analyses.

7.2 Recruitment of Participants

We conducted an extensive study of 40 participants over the web. We used Amazon Mechanical Turk (AMT) to recruit the participants. For the recruitment of workers, we set two criterion. First,



Instructions

You are training an Artificial Intelligence (AI) agent to play Sokoban. Sokoban is a transport puzzle game in which the agent pushes boxes to the goal positions.

This study consists of following phases:

- **Practice Phase**: You control the agent directly. (1-2 minutes)
- **Training 1 - Practice Phase**: You practice how to provide feedback in training 1. (1-2 minutes)
- **Training 1**: You provide positive and negative feedback to an AI agent. (6-7 minutes)
- **Training 2 - Practice Phase**: You practice how to provide feedback in training 2. (1-2 minutes)
- **Training 2**: You tell right action to select to an AI agent. (6-7 minutes)
- **Survey**: You tell us about the usage of the training methods. (<1 minute)

NOTE: For quality purpose, your feedback will be checked against game's ground-truth data.

Fig. 4. Instructions provided to the users on AMT before they accepted to take part in the study.

the worker should have completed and got approved at least 1,000 human intelligence tasks (HITs). Second, out of the HITs completed at least 98% were approved by the HIT requester. We paid 1.25 USD to a worker after he/she completed all the steps of the experiment.

7.3 Web Application Design

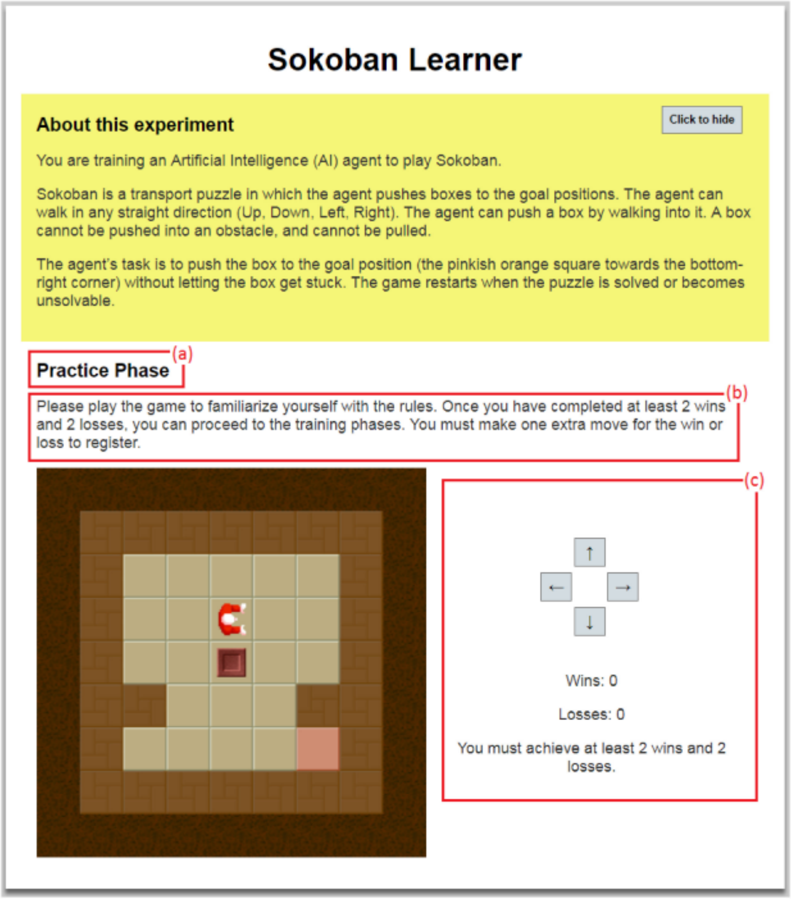
A web interface was designed for the experiments. On AMT, we provided simple instructions to the prospective workers, as shown in Figure 4. There were five phases of interaction with the agent and a survey at the end. Once a worker accepted to work on our task, he/she moved to the experiment's web-app by clicking on the link provided. The landing page of the web-app presented a practice phase, a snapshot of which is shown in Figure 5. The practice was mandatory to make a user familiar with the game of Sokoban. The practice phase required the user to win the game two times as well as lose the game two times. We thought it was a good way of familiarizing the user with the dynamics of the domain.

The practice phase was followed by the training phases for the two methods, SfRA and SfAA. However, before proceeding to a training method, the user was required to go through a practice phase for that method. The practice phase for a training method was intended to panel (a) let the user practice how to use the input panel (b) ensure that the user understands what a correct feedback is. The user needed to provide seven consecutive correct feedbacks, after which a clickable link appeared to move to the main training phase. To make the practice phase consistent for every user, we generated a sequence of 50 state-action pairs, where the actions were chosen randomly. This sequence was repeated in a cycle if required. The user's inputs were matched against the ground-truth labels of actions or rewards, which told the correctness of the feedbacks. The ground-truth consisted of a predefined list of optimal actions from the 50 states or rewards for 49 state-action-state transitions, generated by the first author. For some states two optimal actions were possible; hence, the list contained two optimal action labels or rewards. At the end of the training sessions, the users were asked to respond to a survey questionnaire. Answering the survey was mandatory to be paid. The whole training session took around 20–25 minutes.

The instructions provided to the user for the practice phases and the main training phases are listed in Table 1. The input panels used for the two methods of training are displayed in Figure 6. Note that apart from the bounding-boxes shown in Figure 5, the rest of the items and their positions remained the same in all the phases of interactions. The choice of method 1 or 2 was based on the condition group.

7.4 Algorithmic Settings

Our state representation composed of only x and y coordinates of box and agent, S = $\{(x_1, y_1), (x_1, y_2), \ldots\}$. The action set consisted of A = {left, right, up, down}. We used a start state



Sokoban Learner

About this experiment

You are training an Artificial Intelligence (AI) agent to play Sokoban.

Sokoban is a transport puzzle in which the agent pushes boxes to the goal positions. The agent can walk in any straight direction (Up, Down, Left, Right). The agent can push a box by walking into it. A box cannot be pushed into an obstacle, and cannot be pulled.

The agent's task is to push the box to the goal position (the pinkish orange square towards the bottom-right corner) without letting the box get stuck. The game restarts when the puzzle is solved or becomes unsolvable.

Practice Phase

Please play the game to familiarize yourself with the rules. Once you have completed at least 2 wins and 2 losses, you can proceed to the training phases. You must make one extra move for the win or loss to register.

Wins: 0

Losses: 0

You must achieve at least 2 wins and 2 losses.

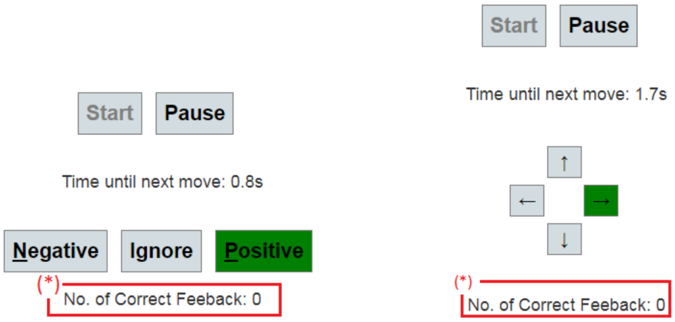
Fig. 5. Snapshot of the web interface designed to conduct experiments with the online users. It show the landing page of the web link. The contents of the place-holder (a) in the subsequent phases was set as either "Practice for Training Phase X" or "Training Phase X." The contents of place-holder (b) are discussed in Table 1. The bounding box (c) contained input panel, which changed according to the training method (shown in Figure 6).

distribution to sample the start state every time an episode started. First, the box was randomly placed in one out of nine free spaces (middle three spaces in each row and in each column). Then, the agent was spawned randomly in one of reaming twenty-two spaces. We set the RL parameters as alpha = 0.1, discount factor = 0.9, and used an ϵ-greedy policy, with $\epsilon = 0.5$. The termination criteria was 200 steps, regardless of the number of the episodes. One reason to limit the study to 200 steps is to avoid the effects of task fatigue, which can occur if the duration of the experiment is beyond the capacity of a common user. This usually results in degradation in the quality of user responses. From some initial runs of the experiment, we found that 2 seconds time-step was sufficient to assign a reward or an action. No user reported this time-duration as a problem in the comments section of the survey during the initial runs. Also, during the main experiments, only one user complaint about the duration of feedback assignment window being small (discussed in detail in Section 9).

Table 1. Showing the Text for the Place-holder (b) as Shown in Figure 5

<table>
<tr><td colspan="2">This practice session is intended to familiarize you with the training technique of phase X.
The AI agent is learning by exploring the maze. It takes an action by itself every 2 seconds. There is no way to directly control the agent's behavior.</td></tr>
<tr><td>Your task is to train the agent by rewarding its action after you see it.
You should give a positive reward (Press "P") if the agent took the right action.
You should give a negative reward (Press "N") if the agent took the wrong action, or got stuck in one place.
You will have 2 seconds to assign the reward before the agent makes its next move.</td><td>Your task is to train the agent by suggesting an action at every time step.
Note: If the agent takes the action that you suggested, then it receives a positive reward. Otherwise, it receives a negative reward.
You should suggest the action by pressing UP (↑), DOWN (↓), LEFT (←), and RIGHT (→) arrow keys.
You will have 2 seconds to provide the action choice before the agent makes its next move.</td></tr>
<tr><td colspan="2">Press "Start" when you are ready to begin.
*NOTE: You must provide at least 7 consecutive correct feedbacks to move to the training phase X.
**NOTE: You must train the agent for at least 200 steps.</td></tr>
</table>

The top and bottom rows show the text common to both methods, SfRA and SfAA. The middle row shows a set of instructions specific for SfRA on left and for SfAA on right. "X" could be 1 or 2. *Instruction particular for the practice phase of a method. **Instruction for the training phase.


Start
Pause
Time until next move: 0.8s
Negative
Ignore
Positive
(*)
No. of Correct Feeback: 0
Start
Pause
Time until next move: 1.7s
↑
←
→
↓
(*)
No. of Correct Feeback: 0

(a)

(b)

Fig. 6. Input panels to record user feedback for SfRA (a) and SfAA (b). The place-holder denoted by (*) was used in the practice phases only.

7.5 Measures

We used various dependent variables to measure the effects of the two training methods (which is our independent variable). First, we used a task performance measure to find out how well the users performed the task of training the agent using the feedbacks. The quality of training is based on the correctness of the feedback. We measured the performance by calculating the proportion of correct feedbacks. Second, we used the following three self-reporting measures,

- First, we measured the "ease-of-use" for the two methods. We wanted to understand if the users perceived one method more natural and easy-to-use than the other method.
- Second, we measured the "mental effort" required to train the agent using a method. It helped in understanding if one method required more mental effort as compared to the other in the view of the users.
- Third, we measured users' preference to use one method over the other for future use. Based on the experiences of using both the methods, which method a user would prefer to use in future?

Note, for a simulation-based domain, there is no suitable scale available to measure the change in user responses based on their experiences with the two different versions of an algorithm. The popular existing scales in human-robot interaction, like, NARS [48], Godspeed questionnaire [7], and the recently introduced RoSAS [9], are mainly concerned with the interaction with a physical robot, which plays a major role in manipulating user's attitude due to embodiment factor [53] and physical presence [35]. However, these measures were not suitable for our study due to the use of a simulated 2D robot.

Besides, we measured the "frustration factor" for both the methods. During the initial (test) runs of the experiments, which were performed with our university colleagues and friends, we observed that the users experienced frustration during training, in case of both the methods. They verbally reported that in SfRA it was frustrating to assign rewards to numerous non-sense actions of the robot. It was due to the necessary exploration by the robot as an attempt to learn a globally optimal policy. Also, in SfAA, it was frustrating to the users when the robot did not obey their commands. For example, if a user previously showed the correct action from the robot's current state, but the robot tried a new action as dictated by its exploration policy, then it caused frustration. We measured these factors separately for both methods.

7.6 Questionnaire Design

After training for 200 steps for both the methods, each user completed a survey questionnaire consisted of seven questions,

Q1: Agent training by positive and negative reward assignment seemed natural and easy-to-use?
Q2: Agent training by action assignment seemed natural and easy-to-use?
Q3: Which method was more demanding in terms of mental effort?
Q4: Which method would you like to use in future to train a robot?
Q5: In reward assignment method, was it frustrating to evaluate agent's actions again and again?
Q6: In action assignment method, was it frustrating when the agent did not obey your action suggestions?
Q7: Any comments?

For the first two and fifth and sixth question, we asked for the ratings. The ratings were provided by the user on a Likert scale, having a response continuum: 1 (strongly disagree), 2 (disagree), 3 (neutral), 4 (agree), and 5 (strongly agree). Note that the ratings were categorical, not numeric. For the third and the fourth question, the user had to choose one option out of four. The last question was an open-ended question and was optional.

7.7 Statistical Tools

To find the statistical significance in the task performance measure, we used student t-test for paired two sample for means. We always used two-tail test results, unless otherwise stated. To analyse the user responses for the survey questions, we compared both within-subject responses and between-subject responses. Within-subject responses were compared when the same user performed experiments with both training methods (i.e., as part of a condition group). Between-subject responses were compared when users' responses for a training method were compared for the two different condition groups. To test significance in within-subject responses (paired samples), we used the Wilcoxon signed rank test [58], and to test significance in between-subject responses, we used the Mann-Whitney test [36]. In addition, to test the correlation between two

Table 2. The First Two Rows Show the Total Number of Learning Steps and the Total Feedbacks Provided

	Condition 1				Condition 2			
	SfRA		SfAA		SfRA		SfAA	
Total steps	3,812		3,882		3,865		3,867	
Total feedbacks	3,184	83.52%	3,184	82.02%	2,885	74.64%	2,916	75.41%
Correct feedbacks	2,627	82.51%	2,853	89.60%	2,352	81.52%	2,562	87.86%
Wrong feedbacks	557	17.49%	331	10.40%	533	18.47%	352	12.07%

The last two rows provide the number of correct and wrong feedbacks and their respective percentages out of the total feedbacks provided. The values are shown for the two conditions for SfRA and SfAA.

variables, we computed Pearson's Product-Moment Correlation Coefficient.[2] And, to test if the proportions of responses for a dependent variable are significantly different, we used Chi-square Goodness of Fit test.[3] Besides, we have provided descriptive analyses using median and mode due to the categorical nature of the responses.

8 RESULTS FOR TASK PERFORMANCE

As mentioned in the previous section, a user trained a policy for only 200 steps. This is a relatively small number of steps to analyze the online and offline performances of a policy (like, rewards per episode, win/loss rate, and steps-to-goal). Therefore, we have analysed the users' performance only in terms of the number of correct feedbacks provided.

In Table 2, we have summarized the data related to the quality of feedbacks. These are the combined results for the 20 users in each case. The total number of steps (row 1) in which the learning occurred was different for SfRA and SfAA in each condition. This difference occurred due to an extra step that was needed at the end of an episode. The extra step is part of a RL algorithm that enables the agent to take an exit action from the terminal state [47]. Therefore, we discarded those steps and ended up with less than 4,000 (20*200) total steps in each case. The difference between max (3,882) and min (3,812) total steps was 70, which shows that the total steps were almost equal in each case. Interestingly, the rate of feedback ($total\ feedback * 100/total\ steps$) in SfRA and SfAA within each condition were almost equal. In condition 1, its mean was ~82.77% (83.52% in SfRA and 82.02% in SfAA), and in condition 2, the mean was ~75.02% (74.64% in SfRA and 75.41% in SfAA). The lower rate of feedback in condition 2 as compared to condition 1 was due to two users who rarely provided feedbacks in both methods, less than ten feedbacks per method.[4] In condition 1, no such user behaviour was found.

The correctness of feedback was decided by looking at the optimality of the feedback. An ideal user (oracle) would always assign optimal actions in case of "action assignment." Moreover, in case of "reward assignment," he/she would always assign positive rewards for optimal actions and negative rewards for non-optimal actions. An optimal action would drive the box to the goal position in minimum steps. We recorded a log file for each user, which contained the information about state before action, state after the action, action, and user feedback. The log file was used offline to compute the rate of correct feedbacks ($correct\ feedbacks * 100/total\ feedbacks$) using the criterion mentioned above. We generated a list containing 506 unique states (unique combinations of the positions of the Sokoban and the box). However, we discarded those states in which the box

[2] As implementation in R: https://stat.ethz.ch/R-manual/R-patched/library/stats/html/cor.test.html.

[3] As implemented in R: https://stat.ethz.ch/R-manual/R-patched/library/stats/html/chisq.test.html.

[4] Inconsistent user behaviour is common in an online labour market [37, 42]. However, we considered survey responses from those users as valid, because they went through all the practice and training phases.

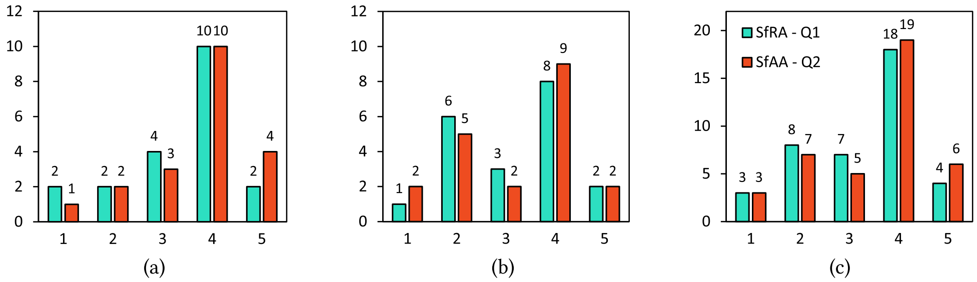


Fig. 7. It shows frequency distributions of responses on the likert sacle. Comparison of if users thought SfRA seemed natural and easy-to-use (Q1) and if users thought SfAA seemed natural and easy-to-use (Q2). In each chart, the x-axis shows the five levels of users' ratings (from 1=Strongly disagree to 5=Strongly agree), and the y-axis represents the frequency of users ratings. (a) Comparison of user responses when SfRA performed first and SfAA second (condition 1). (b) Comparison of user responses when SfAA performed first and SfRA second (condition 2). (c) Combined comparison for both the conditions.

was in a stuck position, i.e., terminal states. This resulted in 242 unique states. The first author, who had years of experience in using the domain, labelled the optimal action(s) from each unique state. There were 104 states with two optimal actions and three states with three optimal actions.

The rate of correct feedbacks was higher in SfAA compared to SfRA in both conditions. In condition 1, 89.60% of the actions assigned were optimal, and 82.51% of the rewards assigned were positive for the optimal actions and negative for the non-optimal actions. The difference between the rate of correct feedbacks was 7.09. In condition 2, 87.86% of the feedbacks in SfAA and 81.52% of the feedbacks in SfRA were correct, with a difference of 6.34. Moreover, we computed the rate of correct feedbacks for each user in both methods. For 20 users in condition 1, the correct feedback rates for SfRA ($\mu = 80.423$, $\sigma = 14.333$) and SfAA ($\mu = 85.696$, $\sigma = 15.769$) were significantly different ($p = 0.015$, $t(19) = -2.652$). In condition 2, the correct feedback rate for SfAA ($\mu = 81.069$, $\sigma = 21.881$) were significantly higher than those of SfRA ($\mu = 68.554$, $\sigma = 33.125$) ($p = 0.029$ as per one-tail t-test results, $t(19) = 2.021$).

9 RESULTS FOR SURVEY QUESTIONNAIRE

In this section, we have presented the survey results for questions Q1 to Q7 as discussed in Section 7.6.

9.1 Ease-of-Use—Results for Q1 and Q2

9.1.1 SfRA vs. SfAA. In this section, we have reported and analysed the users' responses for Q1 and Q2 for both the conditions. Figure 7(a) shows the response frequencies for Q1 and Q2 for users in condition 1, Figure 7(b) shows the response frequencies for the same questions for the users in condition 2, and Figure 7(c) shows the combined response frequencies of the two conditions.

It can be observed from Figure 7(a), 7(b), and 7(c) that at least half of the users agreed or strongly agreed with both methods being natural and easy-to-use. In each case (i.e., condition 1, condition 2, and combined), we found that the response frequencies for Q1 and Q2 closely followed each other, which is reflected by strong and significant correlations between the response frequencies of Q1 and Q2. Moreover, in each condition, one method was not rated more natural and easy-to-use than the other, as the differences between the ratings were not significant. The statistical test results are summarized in Table 3.

Table 3. Statistical Test Results for the Comparison of Responses for Q1 and Q2

	Pearson's correlation coefficient		Wilcoxon signed-ranked test		
	p	R	p	Z	N_same
Q1 & Q2 (Condition 1)	0.017	0.939	0.316	−1.019	8
Q1 & Q2 (Condition 2)	0.014	0.945	0.968	−0.044	9
Q1 & Q2 (Combined)	0.006	0.968	0.465	−0.73	17

N_same is the number of users provided same ratings for Q1 and Q2.

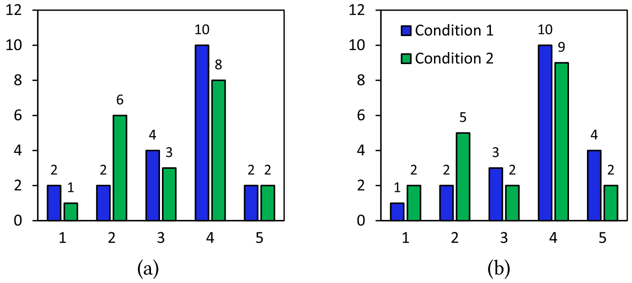


Fig. 8. Comparison of responses for users in condition 1 and for users in condition 2. (a) Comparison for Q1. (b) Comparison for Q2. In each plot, the x-axis shows the five levels of the likert scale and the y-axis shows the frequency of responses. Note, the bars shown in (a) and (b) are identical to the bars shown in Figure 7(a) and 7. These are redrawn here to help readers visually identify the differences and similarities.

Table 4. Statistical Test Results for the Comparison of Responses for Q1 and Q2 in Conditions 1 and 2

	Pearson's correlation coefficient		Mann-Whitney U test	
	p	R	p	U
Condition 1 & 2 (Q1)	0.150	0.742	>0.05	179
Condition 1 & 2 (Q2)	0.084	0.825	>0.05	155

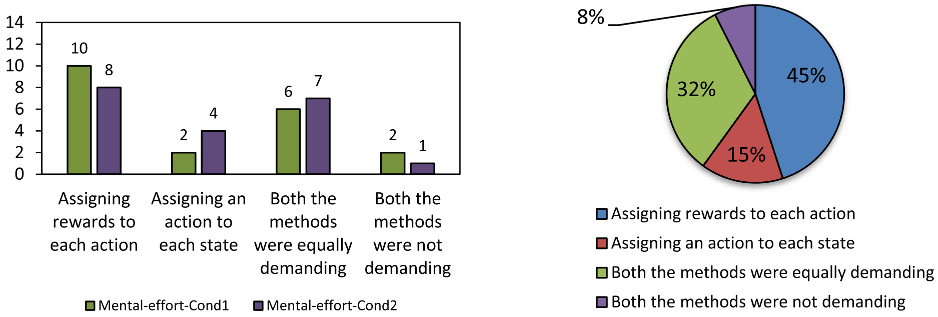
In each Mann-Whitney U test, the critical value of U at $p \leq 0.05$ is 127.

The median and the mode values of responses in both conditions were 4 (i.e., agree), except from the median value in condition 2, which was 3.5.

9.1.2 Condition 1 vs. Condition 2. In this section, we have compared the user responses between the conditions to observe the effect of the two conditions on the user ratings for Q1 and Q2. Figure 8(a) shows the response frequencies on the Likert scale for Q1 for condition 1 and 2. Figure 8(b) shows the response frequencies for the two conditions for Q2.

In both questions, a strong (>0.70) but not significant ($p > 0.05$) correlation existed between the response frequencies in condition 1 and 2. Furthermore, we found that the users did not provide significantly different responses in the two conditions, in case of both Q1 and Q2. Statistically, the conditions did not affect the user responses for the two questions. Table 4 provides a summary of the test results for the comparison of responses for the two conditions.

The only noticeable difference can be observed in the response frequencies for disagree (i.e., rating 2). More users in condition 2 disagreed to SfRA and SfAA being natural and easy-to-use methods than in condition 1.



(a) (b)

Fig. 9. The frequency distribution of the user responses when asked which training method required more mental effort (Q3). (a) Comparison of frequency distribution for the condition 1 and condition 2. The x-axis shows the four options provided to the users and the y-axis shows the frequencies of responses. (b) The pie chart showing the distribution of responses for the four options, combined for both the conditions.

Table 5. Statistical Test Results for the Comparison of Responses for Q3

	Pearson's correlation coefficient		Chi-square Goodness of Fit test		
	p	R	p	df	X
Condition 1	0.119	0.880	0.032	3	8.8
Condition 2			0.111	3	6.0
Combined	-	-	0.003	3	13.8

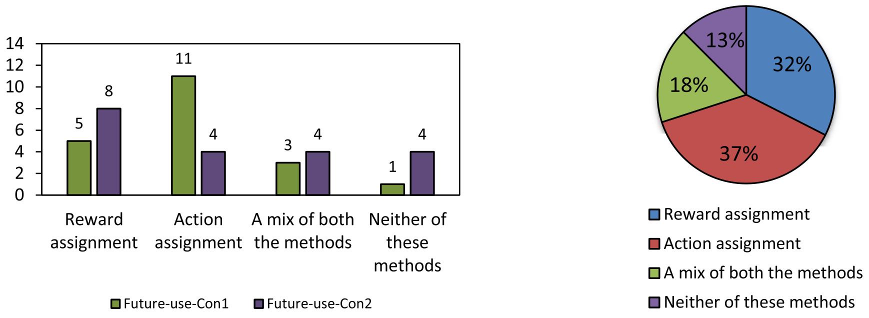
9.2 Mental Effort—Results for Q3

In this section, we will report and analyse the results for Q3, which was to compare the mental effort required in the two methods of training. Figure 9(a) shows the comparison between the two conditions for the response frequencies for Q3. It shows that half of the users in condition 1 and 40% of the users (8 users) in condition 2 thought "reward assignment" required more mental effort than "action assignment." On the contrary, only two users in condition 1 and four users in condition 2 thought "action assignment" was more demanding. A considerable number of users (6 in condition 1 and 7 in condition 2) remained neutral and said both methods required the same effort. Statistical test results showed that the proportions of responses were significantly different in condition 1, but not in condition 2 (see the results in the first two rows of Table 5). Moreover, the results for Pearson's correlation coefficient test in Table 5 shows that the response frequencies in the two conditions were strongly, but not significantly, correlated ($R = 0.880$). The modal response for both conditions was "Assigning rewards to each action."

In Figure 9(b), we have shown the distribution of responses for Q3 combined for both the conditions. Overall, 18 users (45%) thought "reward assignment" was more mentally demanding than "action assignment." and the second highest number of users (13 users or 32%) reported that both the methods were equally demanding. Also, the proportions of responses were significantly different, as shown by the results in Table 5 (last row).

9.3 Future-Use—Results for Q4

In this section, we will report the user responses for Q4, which asked about users' choice for the future use of the training methods. Figure 10(a) shows the comparison of response frequencies for Q4 for the two conditions. It shows that the largest number of users opted "action assignment" in



(a) (b)

Fig. 10. The frequency distribution of the user responses when asked which training method would they prefer to use in future (Q4). (a) Comparison of frequency distribution for the condition 1 and condition 2. The x-axis shows the four options provided to the users and the y-axis shows the frequency of responses. (b) The pie chart shows the distribution of responses for the four options, combined for both the conditions.

Table 6. Statistical Test Results for the Comparison of Responses for Q4

	Pearson's correlation coefficient		Chi-square Goodness of Fit test		
	p	R	p	df	X
Condition 1	1	0	0.010	3	11.2
Condition 2			0.493	3	2.4
Combined	-	-	0.078	3	6.8

condition 1 and "reward assignment" in condition 2. In condition 1, significantly more users opted "action assignment" over reward assignment (Table 6, first row). In condition 2, the largest number of users opted "reward assignment" but the proportions of responses were not significantly different (Table 6, second row). Nevertheless, the combined distribution for the two conditions (Figure 10(b)) shows that the highest number of users (37% or 15 users out of 40) were in favour of using "action assignment" in the future. Moreover, the proportions of responses were not significantly different (Table 6, last row). Furthermore, the response frequencies in condition 1 and 2 did not possess any correlation (Table 6). The mode was "action assignment" in condition 1 and "reward assignment" in condition 2.

9.4 Frustration Factors—Results for Q5 and Q6

In this section, our purpose is to compare the frustration factor for users in condition 1 and in condition 2. Figure 11(a) shows the comparison of frequencies of user responses on the Likert scale when asked if assigning rewards again and again was frustrating. It can be observed that the majority of users agreed or strongly agreed in both the conditions. The differences between responses in condition 1 and condition 2 were not significant. Also, there existed a strong but not significant correlation between the response frequencies. The statistical test results are shown in the first row of Table 7. In both conditions, the mode was 4 (i.e., agree).

The responses for Q6 (Figure 11(b)) show that a vast majority of users in both conditions agreed or strongly agreed that it was frustrating when the agent did not obey the user commands. The modal response was "strongly agree" in both conditions. We found the user responses for the two conditions were not significantly different, and there existed a strong and significant correlation between the response frequencies. The test results are shown in the second row of Table 7.

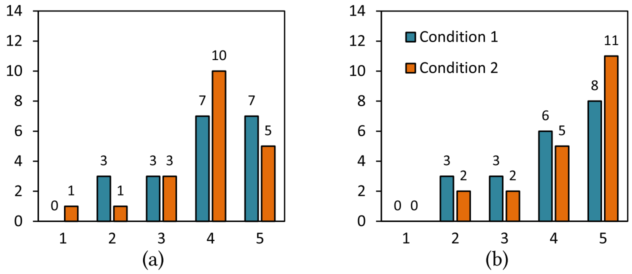


Fig. 11. Comparison of responses between users in condition 1 and in condition 2 for the question (a) if it was frustrating to evaluate agent's actions again and again in SfRA (Q5), and (b) if it was frustrating when the agent did not obey user commands in SfAA (Q6).

Table 7. Statistical Test Results for the Comparison of Responses for Q5 and Q6 in Conditions 1 and 2

	Pearson's correlation coefficient		Mann-Whitney U test	
	p	R	*p*	U
Condition 1 & 2 (Q5)	0.086	0.824	>0.05	192.5
Condition 1 & 2 (Q6)	0.016	0.942	>0.05	167

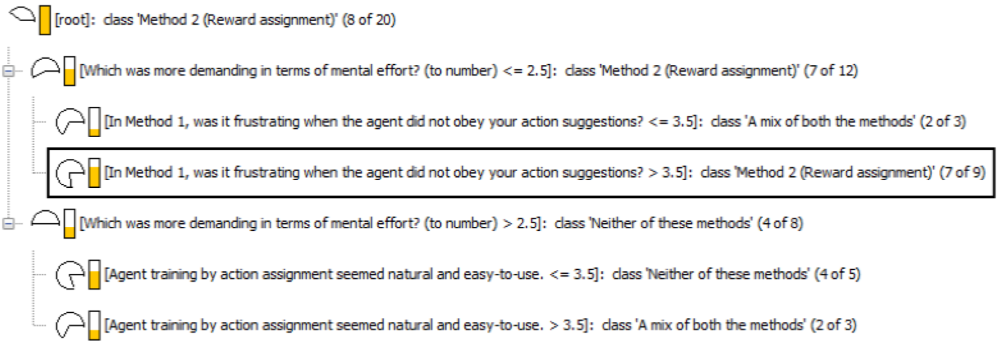
In each Mann-Whitney U test, the critical value of U at $p \leq 0.05$ is 127.

10 DISCUSSION

Task Performance: The results in Section 8 showed that $H1$ holds true. The users had a better idea of what is a correct (optimal) feedback in SfAA as compared to SfRA. In other words, SfAA helped the users in perceiving the optimal actions. We believe it was due to the demonstration-style feedback in SfAA. Moreover, the task performance can be linked to how mentally tough was the task of reward or action assignment. For example, the results in Section 9.2 showed that in general the users found reward assignment more mentally demanding than action assignment. Therefore, it can be the case that the mentally demanding nature of reward assignment resulted in relatively more wrong feedbacks in SfRA. Additionally, it is possible that the two seconds time duration was short for decision making in reward assignment (i.e., the correct reward assignment requires a bit longer than the correct action assignment). However, no user mentioned the shortage of time in SfRA in their (optional) comments.

Although it was not mandatory to provide feedback at every learning step, the overall feedback rate was high for both methods in each condition. It indicates that the users were motivated and wanted to teach the agent most of the times by using their feedbacks. The overall correct feedback rate was also high (above 80%) in each training method. It is an interesting result, because we used optimality condition to decide the correctness of feedbacks, which is a tough criterion. From most of the states, three actions were possible; optimal, suboptimal, and the actions that result in a terminal state. Only optimal actions could lead to the goal using the shortest path and were considered the correct actions. Distinguishing between optimal and suboptimal actions was not an easy task. However, the results in Section 8 suggests that the users could successfully identify a correct action within 2 seconds time-window four out of five times.

Ease-of-use: The results in Section 9.1 indicates that the users in both conditions found the two training methods equally natural and easy-to-use. The users' responses for the two training methods were highly correlated. Therefore, no evidence was found to conclude that one method was better than the other (i.e., $H2$ does not hold true). Instead, the results suggested that both methods



```
[root]: class 'Method 2 (Reward assignment)' (8 of 20)
  [Which was more demanding in terms of mental effort? (to number) <= 2.5]: class 'Method 2 (Reward assignment)' (7 of 12)
    [In Method 1, was it frustrating when the agent did not obey your action suggestions? <= 3.5]: class 'A mix of both the methods' (2 of 3)
    [In Method 1, was it frustrating when the agent did not obey your action suggestions? > 3.5]: class 'Method 2 (Reward assignment)' (7 of 9)
  [Which was more demanding in terms of mental effort? (to number) > 2.5]: class 'Neither of these methods' (4 of 8)
    [Agent training by action assignment seemed natural and easy-to-use. <= 3.5]: class 'Neither of these methods' (4 of 5)
    [Agent training by action assignment seemed natural and easy-to-use. > 3.5]: class 'A mix of both the methods' (2 of 3)
```

Fig. 12. The structure of the decision tree learned using the user responses in condition 2. The rule that discusses the effect of Q3 and Q6 on users preference for reward assignment in Q4 is highlighted by a rectangle.

were significantly similar in providing a natural and usable training method. These are important results, because it shows the proposed interactive feedback method (action assignment) is equally naturalistic to the traditional method of reward assignment, which is seen as a standard method of interactive training in animals and robots (detailed discussion in Section 3). It implies that "action assignment" can substitute "reward assignment" as an interactive feedback method in many applications, without compromising the usability and user-friendliness of the application. Moreover, by overcoming the limitations of SfAA, one of which is reported in Section 9.4, it is possible that the user's acceptance for SfAA may become significantly higher that SfRA.

Mental Effort: The results in Section 9.2 pointed toward the issue that SfRA requires more mental effort due to the assessment of the actions, which was one of our motivations to propose SfAA. There was enough evidence to support our hypothesis that reward assignment requires more mental effort than action assignment ($H3$). A significantly large proportion of overall user responses deemed reward assignment more mentally demanding than action assignment. Also, similar results were observed in both conditions, i.e., the highest number of users said SfRA seemed tougher than SfAA. However, the proportions of responses were not significantly different in condition 2. As mentioned above, the relatively more mental effort required in assigning a reward could have effected the task performance. We believe reward assignment was perceived as mentally demanding, because it involves two cognitive steps, thinking about "what has the agent done" and "what I would have done." On the contrary, "action assignment" involves only one cognitive step, which is thinking "what should I do." However, in a future study, it will be interesting to seek reason(s) from users to find why reward assignment seemed more demanding.

Future Use: The test results in Section 9.3 were inconclusive to support our hypothesis that the users would prefer action assignment over reward assignment for future use; i.e., $H4$ does not hold true. Moreover, the two conditions showed a strong effect on the users' choice for future training method. Interestingly, more than half of the users in condition 1 opted to use action assignment for future use, however, in condition 2, less than one-fourth of the users opted action assignment. There was no apparent reason for why more users selected reward assignment over action assignment in condition 2, therefore, we tried to learn a relation using the responses data. We used a decision tree learner [39] to learn the rules. The learned decision tree is shown in Figure 12. We found, whenever a user said "agree" or "strongly agree" to feeling frustrated when the agent did not obey the action command (Q6), he/she selected "reward assignment" for future use ~78% of times[5] (happened 7 out of 9 times). In this case, it would be possible to decrease the number of users selected "reward assignment" for the future use (and possibly increase the number of users who selected "action assignment" for a future use) by making the agent follow user commands or by decreasing the value of "epsilon" in ϵ-greedy policy.

The overall user responses (combined for the two conditions) showed an interesting trend that the highest number of users opted "action assignment."[6] One reason to include Q4 was to ascertain the quality of user feedback for the first three questions. For example, if significantly more users had found SfAA more natural and easy-to-use and SfRA more mentally demanding, then it would be reasonable to expect that significantly more users prefer SfAA over SfRA for future use. However, it was not the case, because $H2$ does not hold true. Therefore, $H4$ not holding true seems reasonable.

Frustration Factors: From the results in Section 9.4, we found that very high number of users agreed or strongly agreed that both the factors (assigning rewards again and again and the agent not obeying commands) were frustrating. These results highlight that what we found in the initial runs of the experiments (via informal communication) was indeed true. However, the users found agent not following commands in SfAA slightly more frustrating, as reflected by the modal responses for Q5 and Q6, "agree" and "strongly agree," respectively. We note that there could be a psychological factor involved in users getting frustrated when the agent did not obey their commands in SfAA. In SfAA, the demonstration of action is like a direct command to do an action, and if the agent takes a different action than it was obvious that the agent disobeyed the command. However, in SfRA, if a user assigns a reward to action a_1 in state s_n, then the agent revisits state s_n and takes the same action a_1 (as a random action/as part of its exploratory policy). It is likely that the user will overlook that he/she has already provided a reward for this state-action pair. Thus, the agent disobeying the users' previous command may be not that obvious. However, it is possible to overcome the weaknesses pointed out by our results, which can improve the acceptance of interactive reinforcement learning to the end users.

In SfRA, it is possible to ease the load of a single trainer by seeking only a small number of reward feedbacks, as suggested by Raza et al. [40]. In this way, a policy can be taught sequentially by multiple users. In SfAA, one can make agent follow user commands, i.e., act greedily, by using a small value, say 0.1, of epsilon in ϵ-greedy policy. However, acting greedily will compromise the exploration needed to learn a general policy. Moreover, good human-robot interaction design can make learning transparent, resulting in improved user satisfaction [13, 22]. For example, when the agent does not follow the user's suggested action, then it may show a message saying "My policy dictated a different action, but, I have learned from your suggestion." Furthermore, one can adopt an approach similar to Reference [27] in which an agent is forced to use the user's commands for few steps, then, it can return to ϵ-greedy policy.

11 CONCLUSION

This study compared two interactive reinforcement learning methods, training by reward assignment and action assignment, to measure user experiences. It provides many useful findings. First,

[5] We learned a decision tree using the user responses for the first six questions, with Q4 being the class. We used Knime's implementation of the decision tree [8] and used its default parameters, which include the quality measure set to Gini Index and minimum number of records per node set to 2. The learned rule said that when the value of Q3 was less than 2.5 (i.e., user selected either "Assigning an action to each state" or "Assigning rewards to each action") and the value in Q6 was greater than 3.5, then with 77.8% probability the user would select "reward assignment" in Q4.

[6] The p-value was 0.078 for the Chi-square Goodness of Fit test. As pointed out in Reference [50], the "marginal effects" are also important in human-robot studies. They are referred to as "trends," i.e., when $0.05 < p < 0.1$.

we found that feedback as an action assignment enhances a user's capability to perceive the correct behaviour. Second, the proposed method of action assignment seemed natural and comfortable to use to the majority of the users. Moreover, the users rated it equally natural and usable as reward assignment. Third, assigning rewards to agent's actions was declared as a more mentally demanding process than assigning actions. Fourth, there was an overall trend of users preferring "action assignment" over "reward assignment" in case of future use. Fifth, the users in both condition groups found repeated reward assignment a frustrating task. Similarly, they also found it highly frustrating when the agent disobeyed their suggested actions. Based on our findings, we recommend that it may increase the user satisfaction about the learning method if the agent more greedily follows users' feedbacks. Also, it may help if the agent communicates to the user that when it is trying an exploratory action and when it is following the user's command. Note, these results are limited to a specific domain of Sokoban. In future research, it will be interesting to study user responses for more complex, simulated and real-world, robotic domains. Our study has opened prospects for action assignment being a natural and less mentally demanding method of interactive reinforcement learning.

ACKNOWLEDGMENTS

The authors thank Dr. Richard Billingsley, Dr. Sajjad Haider, Dr. Benjamin Johnston, and Jesse Clark for their support.

REFERENCES

[1] Alejandro Agostini, Carme Torras, and Florentin Wörgötter. 2015. Efficient interactive decision-making framework for robotic applications. *Artific. Intell.* 247, C (2015), 187–212.

[2] Tom Anthony, Daniel Polani, and Chrystopher L. Nehaniv. 2014. General self-motivation and strategy identification: Case studies based on Sokoban and Pac-Man. *IEEE Trans. Comput. Intell. AI Games* 6, 1 (2014), 1–17.

[3] Riku Arakawa, Sosuke Kobayashi, Yuya Unno, Yuta Tsuboi, and Shin-ichi Maeda. 2018. DQN-TAMER: Human-in-the-loop reinforcement learning with intractable feedback. *CoRR abs/1810.11748* (2018). *arXiv:1810.11748.* http://arxiv.org/abs/1810.11748.

[4] Brenna D. Argall, Brett Browning, and Manuela Veloso. 2008. Learning robot motion control with demonstration and advice-operators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 399–404.

[5] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robot. Auton. Syst.* 57, 5 (2009), 469–483.

[6] Merwan Barlier, Romain Laroche, and Olivier Pietquin. 2018. Training dialogue systems with human advice. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, 999–1007.

[7] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. 2009. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *Int. J. Soc. Robot.* 1, 1 (2009), 71–81.

[8] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. 2007. KNIME: The Konstanz information miner. In *Studies in Classification, Data Analysis, and Knowledge Organization.* Springer.

[9] Colleen M. Carpinella, Alisa B. Wyman, Michael A. Perez, and Steven J. Stroessner. 2017. The robotic social attributes scale (rosas): Development and validation. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction.* ACM, 254–262.

[10] Sonia Chernova and Andrea L. Thomaz. 2014. Robot learning from human teachers. *Synth. Lect. Artific. Intell. Mach. Learn.* 8, 3 (2014), 1–121.

[11] Francisco Cruz, German I. Parisi, and Stefan Wermter. 2018. Multi-modal feedback for affordance-driven interactive reinforcement learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'18).* IEEE, 1–8.

[12] Francisco Cruz, Johannes Twiefel, Sven Magg, Cornelius Weber, and Stefan Wermter. 2015. Interactive reinforcement learning through speech guidance in a domestic scenario. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'15).* IEEE, 1–8.

[13] M. M. de Graaf and Bertram F. Malle. 2017. How people explain action (and autonomous intelligent systems should too). In *Proceedings of the AAAI Fall Symposium on Artificial Intelligence for Human-Robot Interaction*.

[14] Richard Evans. 2002. Varieties of learning. *AI Game Programming Wisdom* 2 (2002), 15.

[15] Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid Simmons, Kevin Snipes, Alan C. Schultz, et al. 2005. Designing robots for long-term social interaction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*. IEEE, 1338–1343.

[16] Vinicius G. Goecks, Gregory M. Gremillion, Vernon J. Lawhern, John Valasek, and Nicholas R. Waytowich. 2018. Efficiently combining human demonstrations and interventions for safe training of autonomous systems in real-time. *CoRR abs/1810.11545* (2018). *arXiv:1810.11545*. http://arxiv.org/abs/1810.11545.

[17] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L. Isbell, and Andrea L. Thomaz. 2013. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*. MIT Press, 2625–2633.

[18] Kao-Shing Hwang, Jin-Ling Lin, Haobin Shi, and Yu-Ying Chen. 2016. Policy learning with human reinforcement. *Int. J. Fuzzy Syst.* 18, 4 (2016), 618–629.

[19] Charles Isbell, Christian R. Shelton, Michael Kearns, Satinder Singh, and Peter Stone. 2001. A social reinforcement learning agent. In *Proceedings of the 5th International Conference on Autonomous Agents*. ACM, 377–384.

[20] Charles Lee Isbell, Michael Kearns, Dave Kormann, Satinder Singh, and Peter Stone. 2000. Cobot in LambdaMOO: A social statistics agent. In *Proceedings of the AAAI International Conference on Artificial Intelligence (AAAI/IAAI'00)*. 36–41.

[21] Petr Jarušek and Radek Pelánek. 2010. Difficulty rating of sokoban puzzle. In *Proceedings of the 5th Starting AI Researchers' Symposium (STAIRS'10)*. 140–150.

[22] Taemie Kim and Pamela Hinds. 2006. Who should I blame? Effects of autonomy and transparency on attributions in human-robot interaction. In *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication (ROMAN'06)*. IEEE, 80–85.

[23] W. Bradley Knox and Peter Stone. 2009. Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the Fifth International Conference on Knowledge Capture*. ACM, 9–16.

[24] W. Bradley Knox and Peter Stone. 2010. Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 5–12.

[25] W. Bradley Knox and Peter Stone. 2012. Reinforcement learning from simultaneous human and MDP reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 475–482.

[26] W. Bradley Knox, Peter Stone, and Cynthia Breazeal. 2013. Training a robot via human feedback: A case study. In *Proceedings of the International Conference on Social Robotics*. Springer, 460–470.

[27] Samantha Krening. 2018. Newtonian action advice: Integrating human verbal instruction with reinforcement learning. *CoRR abs/1804.05821* (2018). *arXiv:1804.05821*. http://arxiv.org/abs/1804.05821.

[28] Samantha Krening and Karen M. Feigh. 2018. Interaction algorithm effect on human experience with reinforcement learning. *ACM Trans. Hum.-Robot Interact.* 7, 2 (2018), 16.

[29] Samantha Krening, Brent Harrison, Karen M. Feigh, Charles Lee Isbell, Mark Riedl, and Andrea Thomaz. 2017. Learning from explanations using sentiment and advice in RL. *IEEE Trans. Cogn. Dev. Syst.* 9, 1 (2017), 44–55.

[30] Gautam Kunapuli, Phillip Odom, Jude W. Shavlik, and Sriraam Natarajan. 2013. Guiding autonomous agents to better behaviors through human advice. In *Proceedings of the IEEE 13th International Conference on Data Mining (ICDM'13)*. IEEE, 409–418.

[31] Adrián León, Eduardo Morales, Leopoldo Altamirano, and Jaime Ruiz. 2011. Teaching a robot to perform task through imitation and on-line feedback. *Progr. Pattern Recogn., Image Anal., Comput. Vision, Appl.* (2011), 549–556.

[32] L. Adrián León, Ana C. Tenorio, and Eduardo F. Morales. 2013. Human interaction for effective reinforcement learning. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD'13)*.

[33] Guangliang Li, Hayley Hung, Shimon Whiteson, and W. Bradley Knox. 2014. Learning from human reward benefits from socio-competitive feedback. In *Proceedings of the Joint IEEE International Conferences on Development and Learning and Epigenetic Robotics (ICDL-Epirob'14)*. IEEE, 93–100.

[34] Guangliang Li, Shimon Whiteson, W. Bradley Knox, and Hayley Hung. 2017. Social interaction for efficient agent learning from human reward. *Auton. Agents Multi-Agent Syst.* 32, 1 (2017), 1–25.

[35] Jamy Li. 2015. The benefit of being physically present: A survey of experimental works comparing copresent robots, telepresent robots and virtual agents. *Int. J. Hum.-Comput. Studies* 77 (2015), 23–37.

[36] Henry B. Mann and Donald R. Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* 18, 1 (1947), 50–60.

[37] Matthew Marge, Satanjeev Banerjee, and Alexander I. Rudnicky. 2010. Using the Amazon mechanical turk for transcription of spoken language. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP'10)*. IEEE, 5270–5273.

[38] Andrew Y. Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the International Conference on machine Learning (ICML'99)*, Vol. 99. 278–287.

[39] J. Ross Quinlan. 2014. *C4. 5: Programs for Machine Learning*. Elsevier.

[40] Syed Ali Raza, Jesse Clark, and Mary-Anne Williams. 2016. On designing socially acceptable reward shaping. In *Proceedings of the International Conference on Social Robotics*. Springer, 860–869.

[41] Syed Ali Raza, Benjamin Johnston, and Mary-Anne Williams. 2016. Reward from demonstration in interactive reinforcement learning. In *The Twenty-Ninth International Flairs Conference*.

[42] Jon Sprouse. 2011. A validation of Amazon mechanical turk for the collection of acceptability judgments in linguistic theory. *Behav. Res. Methods* 43, 1 (2011), 155–167.

[43] Andrew Stern, Adam Frank, and Ben Resner. 1998. Virtual petz (video session): A hybrid approach to creating autonomous, lifelike dogz and catz. In *Proceedings of the 2nd International Conference on Autonomous Agents*. ACM, 334–335.

[44] Sidney Strauss and Margalit Ziv. 2012. Teaching is a natural cognitive ability for humans. *Mind, Brain Educat.* 6, 4 (2012), 186–196.

[45] Halit Bener Suay and Sonia Chernova. 2011. Effect of human guidance and state space size on interactive reinforcement learning. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (ROMAN'11)*. IEEE, 1–6.

[46] Halit Bener Suay, Russell Toris, and Sonia Chernova. 2012. A practical comparison of three robot learning from demonstration algorithm. *Int. J. Soc. Robot.* 4, 4 (2012), 319–330.

[47] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. Vol. 1. MIT Press, Cambridge.

[48] Dag Sverre Syrdal, Kerstin Dautenhahn, Kheng Lee Koay, and Michael L. Walters. 2009. The negative attitudes towards robots scale and reactions to robot behaviour in a live human-robot interaction study. In *Adaptive and Emergent Behaviour and Complex Systems*. SSAISB.

[49] Ana C. Tenorio-Gonzalez, Eduardo F. Morales, and Luis Villaseñor-Pineda. 2010. Dynamic reward shaping: Training a robot by voice. In *Ibero-American Conference on Artificial Intelligence*. Springer, 483–492.

[50] Andrea Thomaz, Guy Hoffman, Maya Cakmak, et al. 2016. Computational human-robot interaction. *Found. Trends Robot.* 4, 2–3 (2016), 105–223.

[51] Andrea L. Thomaz, Guy Hoffman, and Cynthia Breazeal. 2006. Reinforcement learning with human teachers: Understanding how people want to teach robots. In *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication (ROMAN'06)*. IEEE, 352–357.

[52] Ngo Anh Vien, Wolfgang Ertel, and Tae Choong Chung. 2013. Learning via human feedback in continuous state and action spaces. *Appl. Intell.* 39, 2 (2013), 267–278.

[53] Joshua Wainer, David J. Feil-Seifer, Dylan A. Shell, and Maja J. Mataric. 2007. Embodiment and human-robot interaction: A task-based perspective. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (ROMAN'07)*. IEEE, 872–877.

[54] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. 2017. Deep TAMER: Interactive agent shaping in high-dimensional state spaces. *CoRR abs/1709.10163* (2017). *arXiv:1709.10163*. http://arxiv.org/abs/1709.10163.

[55] Christopher John Cornish Hellaby Watkins. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation. University of Cambridge, England.

[56] Nicholas R. Waytowich, Vinicius G. Goecks, and Vernon J. Lawhern. 2018. Cycle-of-learning for autonomous systems from human interaction. *CoRR abs/1808.09572* (2018). *arXiv:1808.09572*. http://arxiv.org/abs/1808.09572.

[57] Theophane Weber, Sébastien Racanière, David P. Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, Razvan Pascanu, Peter W. Battaglia, David Silver, and Daan Wierstra. 2017. Imagination-Augmented Agents for Deep Reinforcement Learning. *CoRR abs/1707.06203* (2017). *arXiv:1707.06203*. http://arxiv.org/abs/1707.06203.

[58] Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometr. Bull.* 1, 6 (1945), 80–83.

Received January 2019; revised January 2020; accepted May 2020