Abstract: Interactive reinforcement learning (I-RL) is a widely-used technique for agent learning, which allows the agent to receive reward signals not only from the environment but also from a human teacher. In the early stage of I-RL algorithm implementation, prior work usually tests their I-RL algorithms with simulated perfect oracles which are assumed to know the optimal policy for any states and can provide accurate feedback to the learning agent without delay. However, a perfect oracle is not an accurate reflection of human feedback behaviors. To solve this problem, in this paper, we present a participatory design approach that can modify perfect oracles to human-like oracles for I-RL algorithm testing. We conduct a user study using our proposed methodology. We also run experiments with modified oracles generated from our participants. Our modified oracles were found to be more human-like and possibly can provide more realistic testing results compared to perfect oracles.

Keywords: Interactive Reinforcement Learning, Participatory Design

1 Introduction

Interactive Reinforcement Learning (I-RL) is a common technique used in human-agent interaction research, which enables more effective learning for intelligent agents through the use of both environmental observations and additional evaluative information from a human teacher. During development these I-RL algorithms are often tested with simulated perfect oracles in order to obtain preliminary results before conducting final user experiments. These pre-developed perfect oracles already know the optimal policy for any given state and can immediately provide evaluative feedback to a learning agent, and can enable quick iteration during early development of algorithms. However, humans do not behave like perfect oracles. During real human-agent interactions, human teachers could respond to an agent in a delayed, stochastic and unreliable way [1], and could give different feedback in response to the same thing because of their unique personalities, preferences and experience. Therefore, testing results with real humans can differ from those with simulated oracles, which is likely to result in failures during the transition from simulation to real-world testing. Furthermore, this approach fails to incorporate end user perspectives into early algorithm development, which can be a critical problem when research teams do not have adequate representation from the groups who will ultimately use the technologies (e.g. facial recognition technologies not recognizing people with dark skin [2]).

In order to address these challenges, the goal of our work is to develop a methodology that generates more human-like oracles for I-RL algorithms, in collaboration with users. We leverage participatory design methodology and work with participants to build a model of feedback behaviors containing important features that reflect the inconsistencies found between humans and perfect oracles. Then we modify the perfect oracle’s feedback mechanism by integrating the feedback model we created. We did experiments on Lunar Lander simulation environment using baseline Deep Q-Learning (DQN) agent, perfect oracle + Deep-TAMER agent, and several modified oracle + Deep-TAMER agents, separately. The results demonstrated that our modified oracles can reflect human characteristics.

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.
The major contributions of this paper are:

- We introduce a novel participatory design methodology for robotics research communities that involves user perspectives during the I-RL algorithm design process. By constructing human models (personas and feedback models) with participants, designers can rapidly iterate under more realistic testing scenarios;
- We present an idea of exploring human teaching behaviors for agent learning problems. The set of personas and feedback models generated from real people can be used to further understand the relationship between people’s personalities and their teaching styles;
- We alleviate the possible problems caused by the flawed assumption of perfect oracles, and enable the development of more robust I-RL algorithms. For other researchers, the modified oracles with human characteristics can provide multiple functions: 1) allow the agent to learn a policy that better emulates real human’s strategy, 2) bring new opportunities to evaluate how the performance of current algorithm varies within different user types, and 3) serve as a helpful tool to obtain inspirations for improving later iterations.

2 Background and Related Work

Agent learning with human feedback. Interactive Reinforcement Learning (I-RL), formally introduced in [3] as a branch of Reinforcement Learning (RL), allows an agent to interact with not only an environment but additionally a human teacher. Compared to the traditional RL paradigm, I-RL algorithms incorporate a human-in-the-loop to obtain human prior knowledge, and have been proven to be more effective for reducing required training time [4] and improving learning performance [5, 6]. Notably, I-RL can be very useful for some special conditions such as preference-matter tasks [7] and sparse-reward environments [8].

Existing I-RL algorithms typically use human feedback to augment or replace reward functions (e.g. TAMER [9], Deep TAMER [6]), agent policies (e.g. Policy Shaping [10]), and exploration process [11, 12, 13]. However, when applying I-RL algorithms to real-world applications, one gap is that researchers have yet to figure out a good solution to design robust I-RL methods that can be adapted to feedback from different human users, given that variations in human feedback can lead to changes of an I-RL model’s performance [14]. Prior work has found that discrepancy of human feedback can result from many aspects such as frequency [15], delay [16, 17], training strategies [18, 19], randomness [1], and preciseness [20]. Therefore, in this paper, part of our work is to determine what factors play important roles in the human teaching process and explore how these factors work to cause different reactions in the learning agent.

Agent learning with simulated oracles. Apart from real human feedback, feedback produced by simulated human models (oracles) can be used as a learning source for I-RL agents. The idea of using simulated oracles can be traced back to the Oz of Wizard methodology [21] proposed by Steinfeld et al. in 2009, which aims to solve the impracticability of doing user testing at every iteration of new technology development. Later work has proven that introducing simulated oracles is effective for shortening the development cycle and providing useful insights in the early implementation stages [10, 22].

Nevertheless, researchers have also found that results with oracles can not accurately reflect actual results with humans, since simulated oracles are often generated as perfect oracles and overly simplify the human interaction [23]. It is noteworthy that human teachers can actually be expected to perform better than perfect oracles in some cases, for example, when solving problems with multiple solutions [11, 24]. The oracle only knows one of the possible optimal solutions, whereas, humans are able to recognize many other winning policies and thus have a higher chance to achieve the goal.

Participatory design. One of the most common methodologies to understand user behaviors is Participatory Design (PD) [25]. PD incorporates end-users when developing new technologies. It puts an emphasis on integrating researcher’s professional viewpoints with user’s experiences, preferences and expectations, which can reduce the risk of design failure and foster the exploration of new solutions. Recent studies have demonstrated the effectiveness of PD for designing more natural human-robot interactions [26] and evaluating existing artificial intelligence applications [27, 28].
In this paper, we adopt PD methodology in order to understand background differences between individuals and develop human feedback models that can truly reflect people’s personal teaching styles.

3 Methodology

The goals of our work included: 1) understanding divergence of human teaching styles between individuals, and 2) building a model of human feedback that can be used to modify simulated perfect oracles with human characteristics. To achieve these goals, we proposed a participatory design methodology consisting of two one-on-one design workshops. Participants first joined a participatory persona development workshop (Section 3.2) to create a fictional character that can represent their personality and expectations as a teacher. Next, the participant was asked to attend a feedback model design workshop (Section 3.3) to observe and interact with an agent learning environment. University IRB approval was obtained for this process, and participants provided informed consent before the workshops.

Due to local COVID-19 restrictions, all these workshops were designed as an online version using the video conferencing platform Zoom and we were only able to use simulation environments.

3.1 Persona Development Workshop

The persona development workshop aimed to collect and synthesize each participant’s personal background. Under agent learning research settings, the participant worked together with a researcher to generate a fictional user profile (persona) reflecting their behaviors as a human teacher. The personas we created included information from the following four aspects: demographic data, personality, technology experience, and teaching/tutoring experience. These aspects are found to be influential when interacting with robots [29] and associated with evaluations of a learning agent’s behaviors [30]. Other optional items, such as physical and mental states, could be included as well depending on specific research needs.

In our case, since participants had completed a preliminary background survey during the informed consent process, we knew some of their basic information including age range, gender, education level, occupation, area of study, self-identified computer programming level and familiarity with robots beforehand. Then, during the workshop, based on the information we obtained from the survey, we would ask our participants supplementary questions to cover their daily usage of technology products and technical expertise. To scientifically understand participant’s personality, we conducted Big Five Personality test [31] with each participant. To collect data related to teaching experience, we prompted participants to share personal stories and answer some situational questions, for example, “imagine if you are teaching a robot, what kind of teaching style will you use?”.

In addition, participants were be asked to choose an avatar for their persona from a free stock photo library.

![Persona Example](image)

Figure 1: A persona created when conducting the workshop with our participant

Figure 1 is an example of a persona we created with one of our participants. We listed keywords for aspects we believe that would be useful for building persona-behavior mappings in the future. To
Table 1: Attributes of Feedback Model

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Definition</th>
<th>Categorical/Numerical Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>frequency</td>
<td>how often the teacher gives feedback</td>
<td># of timesteps between two feedbacks</td>
</tr>
<tr>
<td>delay</td>
<td>the time that the teacher takes to react</td>
<td>response time</td>
</tr>
<tr>
<td>motivation</td>
<td>whether feedback focuses on positive reinforcement or punishment</td>
<td>reward-focused, punishment-focused, or balanced</td>
</tr>
<tr>
<td>bias</td>
<td>whether feedback is overall more positive or negative</td>
<td>a ratio between 0 and 1</td>
</tr>
<tr>
<td>error</td>
<td>the probability that the teacher gives incorrect feedback</td>
<td>a ratio between 0 and 1</td>
</tr>
</tbody>
</table>

better generalize a user type, we always kept the age of the persona ambiguous, referring to it as a range.

3.2 Feedback Model Design Workshop

The feedback model design workshop sought to understand how different people give evaluative feedback to a learning agent and define the factors causing their discrepancies. Based on the findings in prior research [1, 15, 16, 17, 18, 19, 20], we believe the attributes listed in Table 1 are the most critical ones to be considered when building a feedback model.

In the workshop we held, the participants were asked to operate on a platform, where a discrete Lunar Lander (Figure 2(a)) simulation environment and a UI control interface are shown. An agent in the simulation environment was trying to control a spaceship landing on the area between two flags safely and smoothly. Participants could see the agent’s real-time behaviors and a perfect oracle’s evaluation on the agent’s performance. Evaluative feedback from the oracle is provided as a binary form of either "positive" or "negative", because requesting people to give numerical scores is known to be difficult in reality [32].

The participant first took some time to understand Lunar Lander environment and observe how the current oracle gave feedback to the learning agent. Then, each participant was allowed to change the oracle’s feedback behaviors to emulate their own feedback behaviors. On the UI control interface, feedback frequency, delay time, bias and error could be adjusted through a draggable bar. The participants could also specify whether to use a pure reward-focused or a pure punishment-focused strategy. When the participant was satisfied with current modification, we recorded the parameter value of each feedback attribute.

It is possible that final modification for the oracle still did not reflect the participant’s real feedback behaviors. Therefore, we would also ask the participants to share the ideal oracle’s feedback behaviors they hoped to achieve.

3.3 Testing Framework for Modified Oracles

Figure 3 shows the structure of our testing framework for modified oracles. The testing framework has four major modules: agent, environment, oracle and feedback model.

- **Agent**: an untrained I-RL model which can receive environmental reward and/or evaluative feedback from the oracle.
- **Environment**: a simulation environment using Markov decision process (MDP)
- **Oracle**: a fully-trained RL model (perfect oracle) on the simulation environment, able to provide correct feedback to the learning agent without delay
- **Feedback Model**: a set of feedback behavior attributes (frequency, delay, preciseness, etc.), which can be used to process the oracle’s original feedback

3.4 Perfect Oracle Generation

Perfect oracles used in our experiment are fully-trained DQN models of the LunarLander-v2 environment. We keep training the model until it reaches a successful training condition and then save the current model parameters. Referring to OpenAI Gym Wiki page, the successful training conditions we use for LunarLander-v2 is achieving average reward of 200 over 100 consecutive trials.

For a perfect oracle, one thing we have to define is its reward function. In our experiment, we simply compare the action given by the agent with the action predicted by the oracle. While the learning agent gives an action ($A_{agent}$) based on its current state $s$, the oracle also gives a predictive action

---

Table 2: Modifications to Perfect Oracle

<table>
<thead>
<tr>
<th>Attribute</th>
<th>How to apply modification on the perfect oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>frequency(f)</td>
<td>only generate feedback every f time steps</td>
</tr>
<tr>
<td>delay(d)</td>
<td>provide last feedback generated by the oracle after d time steps</td>
</tr>
<tr>
<td>motivation</td>
<td>reward-focused: skip all negative feedback; punishment-focused: skip all positive feedback; balanced, do not skip any feedback</td>
</tr>
<tr>
<td>bias(b)</td>
<td>get a random probability p, skip negative feedback if ( b &gt; 0.5 ) and ( p &lt; \left( b - 0.5 \right) ); skip positive feedback if ( b &lt; 0.5 ) and ( p &lt; \left( 0.5 - b \right) ); do nothing if ( b = 0.5 )</td>
</tr>
<tr>
<td>error(e)</td>
<td>get a random probability p, inverse feedback if ( p &lt; e ), otherwise keep same</td>
</tr>
</tbody>
</table>

(A\(_{\text{oracle}}\)) based on the same state. If two actions are same \((A_{\text{agent}} == A_{\text{oracle}})\), we interpret the oracle’s feedback as positive feedback with +1 reward. Otherwise, it will be interpreted as negative feedback with -1 reward. When the oracle does not give any feedback, its reward is 0.

3.5 Human-like Oracle Generation and Testing

To construct a human-like oracle, we modified the perfect oracle with the feedback model designed with our participants, as shown in Table 2. For example, if the feedback frequency is \( f \), it means the oracle only gives feedback every \( f \) time steps rather than every time step. To demonstrate the effect of the modified oracles on I-RL, we tested them with Deep TAMER \([6]\), a representative I-RL algorithm. We compared Deep TAMER with the modified oracles to Deep TAMER with a perfect oracle and a DQN baseline, which did not incorporate any human feedback. All algorithms were tested in the Lunar Lander OpenAI Gym environment as described above.

4 Results

4.1 Participants

We collected data from 5 participants (3 males, 2 females), who are all fluent English speakers, currently staying in the United States, aged 18-30 and with diverse academic backgrounds. All but one participant had prior teaching or tutoring experience, and three participants primarily studied or were studying engineering disciplines in college. Two participants believe they have some experience with robots, and four had computer programming experience.

4.2 Teacher Personas

In this section, we describe the personas designed by each participant. The participants’ corresponding modified oracle parameters are shown in Table 3.

Participant 1: Participant 1’s persona was a female undergraduate computer science student who was proficient in programming and had been teacher assistant for many classes before. She described herself as ‘an easygoing and meticulous person’. When constructing her feedback model, she hoped to spend enough time observing the agent’s behaviors so that she could make more accurate evaluation on the agent’s performance.

Participant 2: Participant 2’s persona was a male software development engineer. He had experience of onboarding new employees. He was kind of impatient affected by the fast-paced working environment. This caused his feedback model with higher frequency and also higher probability of giving incorrect feedback. In interpersonal communications, he believed praise is more effective than punishment. However, when it came to agent learning, he felt punishment is also an indispensable part.

Participant 3: Participant 3’s persona was a male graduate student who majors in mechanical engineering. He was self-disciplined and curious about new technology. He had a lot of experience with robots. When setting the value for bias parameter of his feedback model, he believed that maintaining feedback to be slightly overall negative would be helpful as the agent is gradually learning a policy.
Participant 4: Participant 4’s persona was a female medical student. She self-identified herself as ‘a patient and rational person’. She had basic computer programming ability, but was not familiar with robotics. When building the feedback model, she seemed to be a little confused about the simulation environment, and spent a longer time understanding how it works. She believed she would give feedback slowly, because sometimes she was not sure to tell whether the agent’s behavior is desirable or undesirable.

Participant 5: Participant 5’s persona was a male student majoring in business management. He had never formally studied any programming languages nor interacted with robots before. He was interested in technology stuff and played video games quite often. He was quick to understand things and usually felt confident in making decisions.

Table 3: Feedback Models of Each Participant

<table>
<thead>
<tr>
<th>Participant ID</th>
<th>Frequency(step)</th>
<th>Delay(step)</th>
<th>Motivation</th>
<th>Bias(pos vs neg)</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>5</td>
<td>balanced</td>
<td>0.50</td>
<td>5%</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>balanced</td>
<td>0.55</td>
<td>25%</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>3</td>
<td>balanced</td>
<td>0.45</td>
<td>15%</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>4</td>
<td>balanced</td>
<td>0.50</td>
<td>15%</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>4</td>
<td>balanced</td>
<td>0.50</td>
<td>10%</td>
</tr>
</tbody>
</table>

4.3 Learning with Modified Oracles

We tested the performance of Deep-TAMER algorithms with the modified oracles using the testing framework described in Section 3.3. Figure 4 shows the learning curves of baseline DQN agent and Deep-TAMER agents trained with 6 different oracles respectively on Lunar Lander environment.

DQN versus Deep-TAMER: All Deep-TAMER based agents outperform baseline DQN agent on both learning speed and performance, in keeping with prior results [6]. Since Lunar Lander’s reward function is relatively complicated, feedback from oracles helps to guide the agent to learn a policy efficiently and effectively.

Perfect oracle versus modified oracle: Across the five participant-co-designed oracles, there was not a significant discrepancy in learning speed at the beginning of each trial compared with a perfect oracle. However, Deep-TAMER agent trained with a perfect oracle achieves a higher average reward in the fixed number of episode and thus presents better learning performance. In particular, we expect that a learning process with less frequent feedback can degrade the performance of the
algorithm. Furthermore, this demonstrates that testing results with real humans could be different from those with perfect oracles.

**Findings among agents trained with different modified oracles**: We observe that among our five oracles, feedback frequency appears to be the critical factor that determines the agent’s learning performance. Among all other oracles, feedback frequency of modified oracle 4 is the lowest (largest timestep interval). Thus the agent obtained less information from the oracle in the fixed number of training timesteps and thereby had a deteriorated learning performance. That said, frequency is not the only decisive factor. For example, when looking at the learning curve of the agent trained with modified oracle 2, we observe that, even though the oracle gave feedback to the agent most frequently, the agent’s final average reward is still not the highest. This can likely be explained by the relative high error rate of its feedback model, and demonstrates the interplay between the different features of teachers in the context of learning.

5 Discussion and Conclusion

5.1 Qualitative results

When designing feedback models with our participants, all of them choose to use a balanced feedback strategy, providing both positive and negative feedback to the learning agent. However, two participants told us that they would possibly use reward-focused strategy when facing a human learner, for the reason that ‘people might get discouraged if they receive negative feedback’.

We also found that that delay time of all feedback models was similar. All five participants considered delay as their reaction time. In our user study interface, the training time of each step is around 0.1 second, which means our participant’s self-reported reaction time is between 0.3-0.5 second. This is in line with prior results [33] and could be used to parameterize delay in real human-robot interaction scenarios.

5.2 Contributions

The main contribution of our work is that we take advantage of participatory design methodology to enhance the interactive reinforcement learning algorithm development process. By creating personas and feedback models with participants, we are able to better understand the mappings between people’s personalities and their teaching styles, and to ensure that our simulated feedback better reflects real users. We also attempt to generalize a person’s feedback behavior using a parameterized model and then use this model to modify a simulated perfect oracle. The modified oracles with human characteristics are beneficial to the implementation of robust I-RL algorithms: presenting more realistic testing results compared to perfect oracles, and helping researchers come up with more ideas about adapting algorithms to a larger group of users.

5.3 Limitations and future work.

Due to local COVID-19 restrictions, we were only able to conduct the user study remotely with a simulation environment. However, when there are no restrictions, a design workshop with physical robots should be added in the future. To have a more representative participant population, in future work, we plan to include people with more diverse backgrounds. In addition, our feedback model could also be improved. During the user study, we found that feedback attributes such as frequency and error rate are dependent on the simulation environment. Future work is needed to construct a feedback model that can generally represent a human’s behavior regardless of environment changes.

5.4 Conclusion

In this paper, we proposed a participatory design methodology which modifies simulated oracles to be more human-like. We ran a user study with our participants using the proposed method and collected a set of modified oracles. We did experiments with those modified oracles and also the perfect oracle. It is found that our modified oracles are able to show human characteristics and may provide more realistic results compared to the perfect oracle.
References


