Interaction Algorithm Effect on Human Experience with Reinforcement Learning

SAMANTHA KRENING and KAREN M. FEIGH, Georgia Institute of Technology, USA

A goal of interactive machine learning (IML) is to enable people with no specialized training to intuitively teach intelligent agents how to perform tasks. Toward achieving that goal, we are studying how the design of the interaction method for a Bayesian Q-Learning algorithm impacts aspects of the human's experience of teaching the agent using human-centric metrics such as frustration in addition to traditional ML performance metrics. This study investigated two methods of natural language instruction: critique and action advice. We conducted a human-in-the-loop experiment in which people trained two agents with different teaching methods but, unknown to each participant, the same underlying reinforcement learning algorithm. The results show an agent that learns from action advice creates a better user experience compared to an agent that learns from binary critique in terms of frustration, perceived performance, transparency, immediacy, and perceived intelligence. We identified nine main characteristics of an IML algorithm's design that impact the human's experience with the agent, including using human instructions about the future, compliance with input, empowerment, transparency, immediacy, a deterministic interaction, the complexity of the instructions, accuracy of the speech recognition software, and the robust and flexible nature of the interaction algorithm.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**; **Interaction design**; *Interaction techniques*; • **Computing methodologies** → Reinforcement learning; Learning from critiques;

Additional Key Words and Phrases: Human-agent interaction, reinforcement learning, human factors, natural language interface, sentiment

ACM Reference format:
Samantha Krening and Karen M. Feigh. 2018. Interaction Algorithm Effect on Human Experience with Reinforcement Learning. *ACM Trans. Hum.-Robot Interact.* 7, 2, Article 16 (October 2018), 22 pages.
https://doi.org/10.1145/3277904

1 INTRODUCTION

As part of the larger field of interactive machine learning (IML), this work aims to understand how to create intelligent agents that can easily be taught by individuals with no specialized training, using an intuitive, interactive teaching method such as critique or demonstrations. The widespread integration of robotics into everyday life requires significant improvement in our understanding of what makes machine learning (ML) agents intuitive and pleasurable for individuals who are not ML experts to interact with. Instead of asking how to train a person to use intelligent agents, this research asks how to design agents so they can be easily trained by people.

Authors' addresses: S. Krening, Georgia Institute of Technology, 270 Ferst Drive, Atlanta, GA, 30332; email: skrening@gatech.edu; K. M. Feigh, Georgia Institute of Technology, Atlanta, GA, 30332; email: karen.feigh@gatech.edu.

2573-9522/2018/10-ART16
https://doi.org/10.1145/3277904

This article focuses on the question: *Does the interaction method affect the person's experience with the agent, and if so, why?* Specifically, we analyze whether the method of natural language feedback given to an ML agent impacts the human teacher's satisfaction. Human experiences are important because individuals who experience frustration while interacting with a robot are unlikely to interact with the robot in the future (Parasuraman and Riley 1997).

Traditional research into IML often fails to account for the human experience. Instead, algorithms are tested with computer simulations of human input, i.e., oracles, and verified using objective ML metrics, such as cumulative reward. While a good starting point, this method should be considered incomplete for any system that is intended to learn from humans, as it ignores important aspects of the human-agent interaction that are difficult to gauge using oracles alone. The use of oracles instead of human-subject experiments overly simplifies the human interaction and ignores how humans react to the agent. Oracles will never get frustrated with the agent or confused by its actions. We suggest that researchers should design and evaluate IML algorithms using human subjects and human factor metrics for testing and verification in addition to the traditional ML analyses in order to measure and improve the human teacher's experience.

We performed a repeated measures human-subjects experiment in which participants taught two agents how to play a simple game with different teaching methods: critique and action advice. For the first teaching method, people trained the agent using positive and negative verbal critique, such as "good job," and "don't do that." The second teaching method enabled people to teach the agent using action advice, such as "move right" and "go left." The same underlying machine-learning algorithm, Bayesian Q-Learning, was used for both agents. The critique agent used Policy Shaping (Griffith et al. 2013) as an interaction algorithm, while the action advice agent used the Newtonian Action Advice algorithm (Krening 2018).

Participants had a much better experience with the action advice agent compared to the critique agent. Participants found the action advice agent to be more intelligent, less frustrating, clearer, and more immediate in terms of how the agent used human input, was better able to complete the task as the person intended, and was more intelligent than the critique agent. Compared to the critique agent, the advice agent had a shorter average training time, a smaller number of steps to complete each episode, and a higher average reward. People provided approximately the same amount of instruction for both the advice and critique agents.

2 BACKGROUND

2.1 Reinforcement Learning

Reinforcement learning (RL) is a class of machine-learning algorithms in which intelligent agents learn what actions to take by receiving a signal of rewards and punishments from the environment (Sutton and Barto 1998). Like other works in IML (Sahni et al. 2016; Subramanian et al. 2016), this work incorporates human instruction into an RL agent; however, we focus on the human teacher's experience.

For example, one of the aspects of the human experience we are investigating is whether the interaction algorithm affects how intelligent human teachers perceived the agent to be. Many studies show that how people perceive the intelligence of a person or robot is impacted by other factors of interaction. A robot's physical appearance affects how intelligent people expect the robot to be DiSalvo et al. (2002), Duffy (2003), and Fong et al. (2003). The degree that a robot's behavior is life-like is strongly correlated to its perceived intelligence (Bartneck et al. 2009). A person's accent affects how intelligent that person is perceived to be Phillips (2010), Rakić et al. (2011). We expect that the perception of the agent's intelligence is affected by the type of teaching method.

RL is influenced by behavioral psychology (Skinner 1990; Sutton and Barto 1998). B.F. Skinner wrote about "selection by consequences," comparing the evolution of living things through

natural selection with the shaping of individual behavior through reinforcement (Skinner 1981). The probability people will repeat an action in a given circumstance is increased or decreased if they receive positive or negative reinforcement. Since one way people learn is by interacting with their environment, we chose to mirror this method when choosing a machine-learning algorithm to teach our agent. Unlike the Inverse Reinforcement Learning problem that generates a reward function based on an ideally comprehensive set of optimal behavior (Ng et al. 2000), in this work a reward signal is given along with real-time human instructions to enable an RL agent to converge to an optimal or near-optimal policy in less time than if no human input were provided.

Most RL algorithms are modeled as Markov Decision Processes (MDPs), which learn policies by mapping states to actions such that the agent's expected reward is maximized. An MDP is a tuple (S, A, T, R, γ) that describes S, the states of the domain; A, the actions the agent can take; T, the transition dynamics describing the probability that a new state will be reached given the current state and action; R, the reward earned by the agent; and γ, a discount factor in which $0 \leq \gamma \leq 1$.

Bayesian Q-Learning was the (MDP-based) underlying RL agent used for both the critique and action advice conditions in the experiment. Bayesian Q-Learning is an RL algorithm in which the utility of state-action pairs is represented as probabilistic point estimates of the expected long-term discounted reward (Dearden et al. 1998).

2.1.1 Learning from Critique. We chose the Policy Shaping algorithm for the agent that learns from critique in this experiment. Policy Shaping is an interaction algorithm that enables a human teacher's critique to be incorporated into a Bayesian Q-learning agent as policy information (Griffith et al. 2013).

Historically, critique was used directly as a reward signal (Isbell et al. 2001), but this has certain issues including the fact that people stop providing critique once the agent learns the task, which makes critique an unpredictable signal (Isbell et al. 2006). It was later shown (Knox and Stone 2010; Thomaz and Breazeal 2008) that it is more efficient to use critique as policy information. Policy Shaping is an interaction algorithm that enables a human teacher's critique to be incorporated into a Bayesian Q-learning agent as policy information (Griffith et al. 2013) and was used in this work. Cederborg et al. (2015) investigated how to interpret silence while learning from critique with Policy Shaping.

There are some basic behavioral templates that people expect in a teaching situation. For example, if I were to tell my nephew, "Stop," I would expect him to immediately change his behavior. Algorithms like Policy Shaping, which more efficiently use critique as policy information instead of directly as a reward signal, do not adhere to this teaching template. This work analyzes whether this causes people to be confused and frustrated with the agent.

2.1.2 Learning from Action Advice. For the agent that learns from advice in this experiment, we created an algorithm called Newtonian Action Advice (Krening 2018). The interaction is based on Newtonian dynamics: objects in motion stay in motion unless acted on by an external force. In the Newtonian Action Advice (NAA) model, action advice provided by the human is an external force on the agent. Once a person provides action advice (e.g., "Move left"), the agent will immediately move in the direction of the external force, superseding the RL agent's normal action selection. The model contains natural friction that "slows down" the agent's need to follow the human's advice. The friction ensures that after some amount of time, the agent will resume the underlying RL algorithm's action selection policy. For example, if the teacher provides advice to move left and the model's friction parameter is set to five steps, the agent will deterministically follow the advice and move left for five time steps. After five time steps have elapsed, the agent will return to its action selection method.

Similar to Policy Shaping, the Newtonian Action Advice agent also used Bayesian Q-learning as the underlying RL algorithm. Bayesian Q-learning was chosen as the RL algorithm for Newtonian Action Advice in order to make both agents (Policy Shaping and NAA) more similar and directly comparable; with no human input, both agents are reduced to a Bayesian Q-learning agent and have identical performance. The NAA algorithm can use a different underlying RL algorithm if desired, but the Policy Shaping algorithm cannot use anything other than Bayesian Q-learning without fundamental algorithmic modifications.

Various forms of advice have been developed in other work, including linking one condition to each action (Maclin et al. 2005) and linking a condition to rewards (MacGlashan et al. 2015). Several connect conditions to higher-level actions that are defined by the researcher instead of primitive actions (Joshi et al. 2012; Kuhlmann et al. 2004; Maclin et al. 2005). Argall et al. (2008) creates policies using demonstrations and advice. Meriçli et al. (2014) parses language into a graphical representation and finally to primitive actions. Maclin et al. (2005) has the person provide a relative preference of actions, whereas the agent determines the order of preferred actions in our work. Sivamurugan and Ravindran (2012) explored learning multiple interpretations of instructions. Tellex et al. (2011) represents natural language commands as probabilistic graphical models.

Most other forms of advice algorithms are permanently influenced by the advice (Kuhlmann et al. 2004; Maclin et al. 2005). Newtonian Action Advice differs because the advice can be overwritten by new, contradictory advice in the future.

2.2 Natural Language Processing

2.2.1 Automatic Speech Recognition (ASR). ASR is how the human's verbal instructions are transcribed to written text. This experiment used the Sphinx ASR software (Huggins-Daines et al. 2006).

2.2.2 Sentiment Analysis. Sentiment analysis is a natural language processing (NLP) tool that has traditionally been used to classify book, movie, and product reviews into positive and negative (Pang and Lee 2008). Sentiment analysis has not been widely used for action selection. One method we previously developed for using sentiment analysis is to classify natural language advice into advice of "what to do" and warnings of "what not to do" (Krening et al. 2017). Many approaches to learning from language instruction require people to provide instructions using specific words, often in a specific order or format (Meriçli et al. 2014). Thomason et al. (2016, 2015) worked to get around some of these limitations like keyword search by creating an agent that learns semantic meaning from the human and learning the grounding of natural language descriptions of objects using multimodal sensory perception. In this work, we develop a method of using sentiment analysis to filter verbal critique into positive and negative, which furthers the goal of allowing people to provide verbal instructions without being limited to a specific dictionary of words.

This work uses Stanford's deep learning sentiment analysis software (Manning et al. 2014), which uses Recursive Neural Tensor Networks and the Stanford Sentiment Treebank (Socher et al. 2013). The sentiment analysis software classifies input text as binary positive and negative, not a number indicating magnitude, and has an accuracy around 80% to 86%. The Stanford Sentiment Treebank is a set of labeled data corpus of fully labeled parse trees trained on the dataset of movie reviews from rottentomatoes.com (Pang and Lee 2005).

3 CRITIQUE VERSUS ACTION ADVICE

This article focuses on the question: *Does the interaction method affect the person's experience with the agent, and if so, why?* The first goal is to determine whether the interaction algorithm can affect the human teacher's experience. To support that goal, this work compares two varied interaction

algorithms: Policy Shaping that learns from critique, and NAA that learns from action advice. The second goal is to analyze the participants' responses to explore *why* the interaction algorithm impacts the human's experience. Once we have determined which design considerations affect the person's relationship with the agent, we can (1) design experiments in future work to isolate and test the impact of each characteristic and (2) create interactive ML algorithms that adhere to the design characteristics that foster a positive relationship between the human teacher and ML agent.

To that end, in this section we discuss several differences between advice and critique that we expect to impact the human's experience, including rhetoric, timing, and whether the instructions apply to the future or past.

3.1 Rhetoric

We suggest that the rhetoric of critique is inherently negative, while advice is more positive. Since action advice exclusively informs the agent about what to do next, there is a sense of rhetorical forgiveness– the human does not judge the agent's behavior, but rather offers a helping hand and advises the agent what to do next to move to a better situation. Critique, on the other hand, is about placing credit and blame on past actions without encouraging or allowing a person to provide a potential solution of what to do next (Heinrichs 2017).

Suppose that the agent was driving a car and got stuck in the mud. Rhetorically, the critique teaching method is like a passenger crossing his arms, glaring at the driver, and saying, "You are a terrible driver." This may be true, but it does not help the situation. The action advice teaching method would be like a passenger who, without judging, suggested using twigs, leaves, or a car mat to create traction and told the driver to slowly accelerate out of the rut.

The inherent rhetoric of different teaching methods has never been applied to IML algorithm design before, but we believe it is a characteristic that can impact the human experience. We do not want to design algorithms that force human teachers to be the judgmental backseat driver in perpetuity as this would likely result in endless frustration. We believe in creating a more positive interaction between the agent and human teacher.

3.2 Immediate Applicability of Feedback

Critique provides rewards or punishments about past behavior, while advice supplies actions that should be taken by the agent in the immediate future. The critique will only affect the Policy Shaping agent's behavior if the critiqued past state is revisited in the future. Because the critique agent applies feedback to past instead of future actions, there is no way to "close the loop" so the human can immediately tell if his or her feedback was correctly understood and applied. Action Advice, on the other hand, provides instructions that the agent can immediately apply. Consequently, it will likely seem to the human that the action advice agent is more responsive than the agent learning from critique. Additionally, people give instructions about the future, even if expressly told not to Thomaz and Breazeal (2008). In these terms, we expect action advice to be more human-friendly than critique.

3.3 Issues of Language Processing

There are three main issues associated with language processing when using verbal instruction as an interaction medium that impact the advice and critique algorithms unequally: language processing time, accuracy of the ASR software, and credit attribution. The advice agent suffers more with the language processing time and accuracy of the transcription software, while critique is impaired by credit attribution.

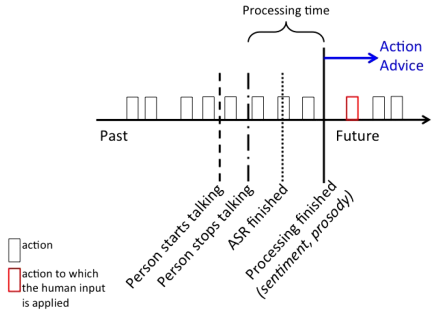

Fig. 1. Advice applies to the next action after language processing.

(1) With the action advice agent, people are sensitive to the *language processing time* because the agent's next action is affected. Human teachers are aware of the processing time because it acts as a lag between when the person gives advice and when the agent follows the advice. A long processing time would degrade the human experience. With the critique agent, however, people are unaware of the processing time because their instructions are used to inform past behavior.
(2) Human teachers are expected to be more sensitive to the *accuracy of the automatic speech recognition software* with the action advice agent because the human can immediately tell if the agent understood based on the next action. People are unaware of the speech software accuracy with the critique agent because the next actions are not impacted.
(3) One severe issue with critique is the *credit attribution problem*. There is no definitive way to know which past action(s) the human teacher intended to reward or penalize. Action advice does not have this problem because the human's action instructions are simply applied to the next action. The credit attribution problem may be more apparent to the algorithm designer than the human teacher–the human will know the critique agent is not performing well but will not necessarily know if, and how much, credit attribution is to blame.

Figure 1 shows how language processing impacts the advice agent. At some point in time, the human teacher starts giving verbal action advice, such as "Move left." After the person stops speaking, the ASR software transcribes the speech to text. Additional language processing, such as sentiment analysis, may be performed on the text depending on the agent setup. After language processing, the advice is applied to the next action in time. The human teacher can sense longer language processing times since that would cause the agent to not follow the advice for several actions. Assuming a relatively uniform time step, people soon learn the lag in the system. People are also sensitive to the ASR accuracy because they will immediately tell if the verbal instruction was incorrectly transcribed when the agent does not follow their advice. Significant inaccuracies in speech recognition degrade the human experience. Finally, there is no credit attribution problem with an advice agent.

Figure 2 shows how language processing impacts the critique agent. At some point in time, the human teacher starts to give verbal critique, such as "Good job." Once the person stops speaking, the language is processed. After the language is processed into positive and negative critique, the critique is used to update the Policy Shaping agent. The human teacher is not aware of either the language processing time or the ASR accuracy since critique is applied to past actions instead of the future. The human teacher is also not necessarily aware of the credit attribution problem, even though it is a severe issue with critique. It may appear at first glance that the three issues of

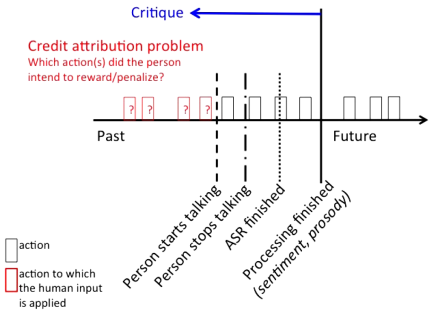

Fig. 2. The credit attribution problem with critique.

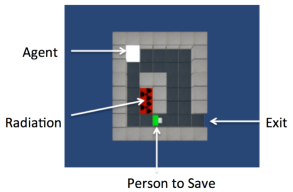

Fig. 3. Radiation World initial condition.

language processing do not impact the human's experience of the critique agent as much as the advice agent. However, the teacher does not know if (1) the critique has been heard, (2) the critique has been correctly interpreted, and (3) to which action(s) the critique has been applied.

4 METHOD

We conducted a repeated-measures human-subject experiment in which we investigated the effect of two different teaching methods, critique and action advice, on the human's satisfaction with the agent. The experimental setup has been previously described in Krening and Feigh (2018), which details a small subset of the results presented in this work. The experiment collected data from 24 participants who were not experts in machine learning, and in many cases were not associated with a university. The age of the participants varied from 18 to 65 years old.

Each participant trained two agents with different interaction methods–critique and action advice–but an identical underlying Bayesian Q-learning algorithm. The experiment split participants into two groups randomly. The first group trained the critique agent first, and the second group trained the advice agent first.

The task required participants to teach each agent to rescue a person in Radiation World, which is a simple game developed in the Unity Minecraft environment as shown in Figure 3. In the experimental scenario, there has been a radiation leak and an injured person is located somewhere in the grid unable to move. The agent must find and rescue the person and take him to the exit, all while avoiding the radiation. The task is complete if the agent takes the person to the exit. The task is failed if the agent enters the radiation or exits without the person. The task is repeated for several training episodes; participants were told to stop training when either the agent was performing as they intended or the participant was too frustrated to continue or wanted to stop for any reason. Consequently, the training time varied for each participant and teaching method. After a participant finished training an agent, the participant completed a questionnaire concerning the experience (*see Paired Tests*). After training both agents, the participants filled out a questionnaire comparing the two agents (*see One-Sample Tests*). At the end of the experiment, participants

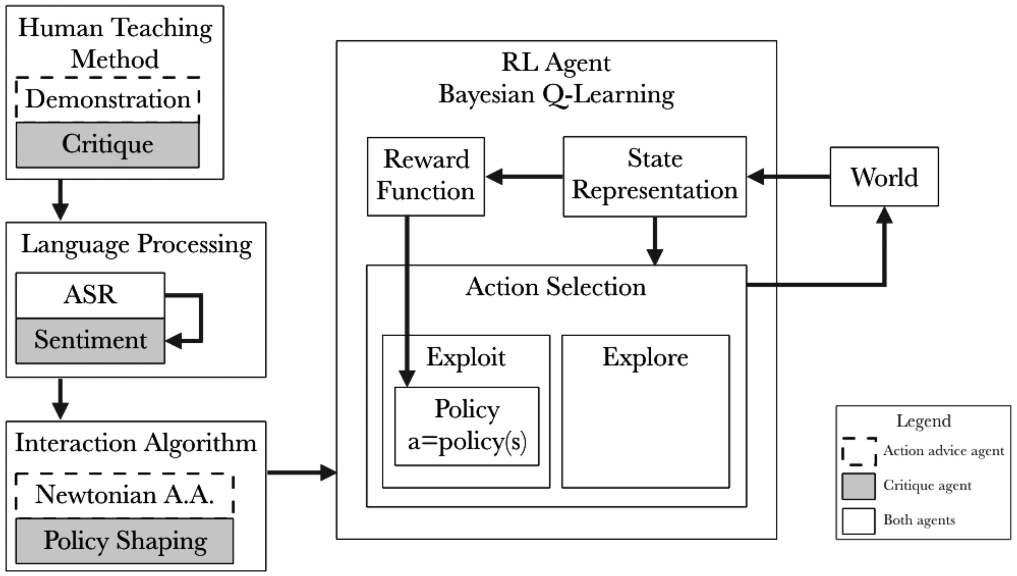

Fig. 4. Agent setup.

were asked if they had any questions or additional thoughts concerning the experiment in an exit interview.

The agent setup is shown in Figure 4. Both agents learned from verbal instruction. The action advice agent used what can be thought of as verbal demonstrations, while the critique agent learned from positive and negative critique. The verbal instructions were transcribed to text using ASR software. The critique agent had an additional language processing step of using sentiment analysis to convert the text to positive or negative critique. After language processing, the processed instructions were sent to an interaction algorithm. The advice agent used Newtonian Action Advice, while the critique agent used Policy Shaping. Both interaction algorithms worked with the Bayesian Q-learning algorithm to determine action selection.

Neither agent is guaranteed to perform better than the other. If the human provides no instructions, the critique and advice agents perform equally. With human teachers, however, the performance of each agent is entirely dependent on the instruction provided by the person. In terms of theoretical algorithmic performance for an equivalent set of human instructions, the Newtonian Action Advice agent with a friction parameter of $S = 5$ *steps* (advice followed for five time steps immediately after the advice is given) in this experimental scenario is likely to have a higher cumulative reward for the initial episodes, but the Policy Shaping agent is likely to earn a higher cumulative reward for the later episodes (Krening 2018).

To teach the critique agent, participants were instructed to provide positive or negative critique if the agent did something they considered good or bad. Using sentiment analysis as a filter allowed people to provide verbal natural language critique without restricting their vocabulary. For example, a participant could give varied critique such as, “Good job,” “That’s great,” “That is a bad idea,” and “You’re wasting time.” The critique agent used Policy Shaping to incorporate the positive and negative feedback.

For the action advice agent, people were instructed to tell the agent to move in a desired direction. For example, if participants wanted the agent to move down, they should say “down.” The only four words recognized by the action advice agent were “up,” “down,” “left,” and “right.” The

action advice agent used the Newtonian Action Advice algorithm to incorporate the advice with the Bayesian Q-learning algorithm's action selection. The friction value used in the experiment was $S = 5(steps)$.

This experiment is designed to determine if the interaction algorithm impacts the human's satisfaction with the agent. Specifically, we analyze whether the Newtonian Action Advice agent creates a better experience for the human than Policy Shaping. This experiment is not designed to guarantee that all action advice algorithms will create a better user experience than all critique algorithms. However, the exploratory part of this experiment asks participants why one interaction algorithm impacted different aspects of their experience such as frustration. We plan to analyze these participant explanations in order to learn which algorithmic design characteristics may enhance or detract from the human's experience. Once these design characteristics are identified, we can analyze each independently and in greater depth in future work.

Experimental Procedure

(1) Greeting and introduction.
(2) Instructions for agent 1. Train agent 1. Questionnaire about experience of training agent 1. *See Paired Tests in the Results.*
(3) Instructions for agent 2. Train agent 2. Questionnaire about experience of training agent 2. *See Paired Tests in the Results.*
(4) Questionnaire comparing the experiences of training both agents, including free-response questions. *See One-Sample Tests and participant quotes in the Results.*
(5) Exit interview.

4.1 Human Experience Measures

The Human Experience measures were created as a modified version of the NASA-TLX questionnaire (Hart 2006). The answers to all questions regarding human experience were collected using a sliding bar in which the selected value to the tenths decimal place was shown to the participant.

Paired Tests. Immediately after training an agent, the participants were asked to score the intelligence of the agent on a continuous scale from [0:10]. A value of 0 indicated that the agent was not intelligent, while 10 meant very intelligent. The same scale of [0:10] was used for four additional metrics: performance, frustration, transparency, and immediacy. Values of 0 corresponded to poor performance, low frustration, nontransparent use of feedback, and a slower response time. Values of 10 meant excellent performance, high frustration, clear use of feedback, and an immediate response time, respectively.

One-Sample Tests. After training both agents, participants were asked to compare the intelligence of the two agents on a continuous scale from [-5:5]. A value of -5 indicated that the action advice agent was much less intelligent than critique. Zero meant the two agents were equally intelligent. A value of 5 meant the action advice agent was much more intelligent than critique. The same scale of [-5:5] was used for performance, frustration, transparency, and immediacy. Values of -5 corresponded to the advice agent performing much worse, causing much less frustration, using feedback less clearly, and responding much slower than critique. The scales are shown on Figures 6 through 10.

Explanation of Responses. After training both agents, in item 4 of the procedure participants were given the option to explain the answers that they gave on each of their five comparison ratings. For example, participants were asked, "If one agent was more frustrating than another, what made you feel that way?" These written responses were entirely free form with no priming or options provided by the experimenter. The purpose of these explanations is to explore *why* the interaction algorithm impacts the human's experience. The resultant set of design characteristics that

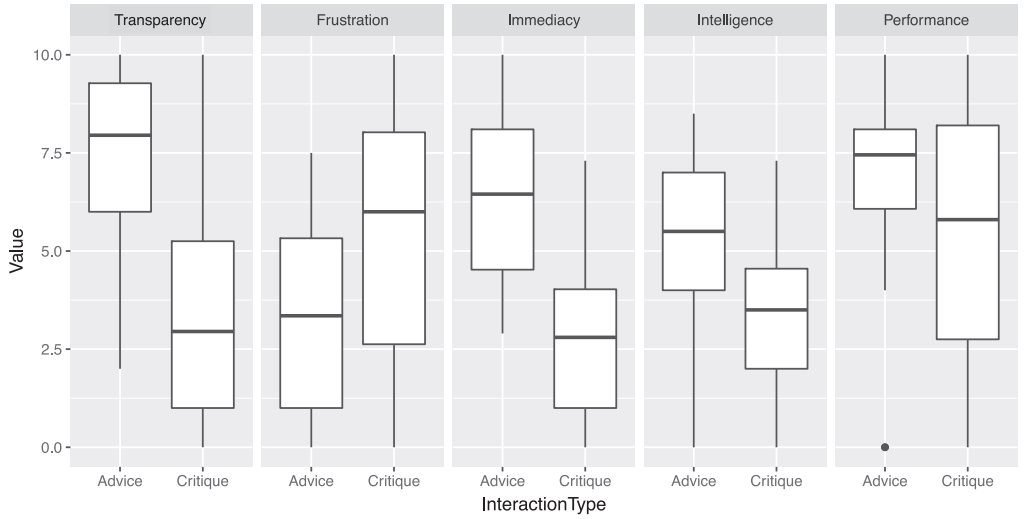

Fig. 5. Human factor metrics. For all metrics save frustration, higher values represent a better human experience (better performance, greater transparency, more immediate, and more intelligent). For frustration, higher values indicate a higher level of frustration, and therefore a worse human experience.

affect the person's relationship with the ML agent allows us to analyze each of these characteristics thoroughly in future work and eventually use the characteristics to direct the design of IML algorithms.

4.2 Objective ML Measures

While participants were training the agents, objective performance metrics were logged to data files. The training time and reward earned per episode were measured as continuous data. The amount of human input provided and the number of actions it took to complete an episode were measured as ordinal count data.

5 RESULTS

We will first show the results of the human teachers' experience interacting with the agents, followed by an analysis of the objective metrics of the agents' performances. For the human factors metrics, we will begin with an overview of the quantitative analysis and then provide more detailed discussion of each metric individually.

5.1 Human Experience Overview

Figure 5 illustrates the results of how participants scored aspects of the human experience. The differences indicate that the human experience was not the same for the advice and critique agents. The figure shows aggregate participant ratings of how frustrated they were with each agent, whether it was clear how each agent used the human instruction, how quickly each agent responded to the instruction, whether the agent performed the task as intended, and how intelligent the person thought the agent was.

Paired Tests on Human Experience. Paired t-tests were conducted for each of the five human factors metrics that were collected following training each agent type. The null hypothesis was that the pairwise difference between the two paired groups had a mean equal to zero. A significance of $\alpha = 0.05$ was chosen as a reasonable limit on Type I error given the number of participants

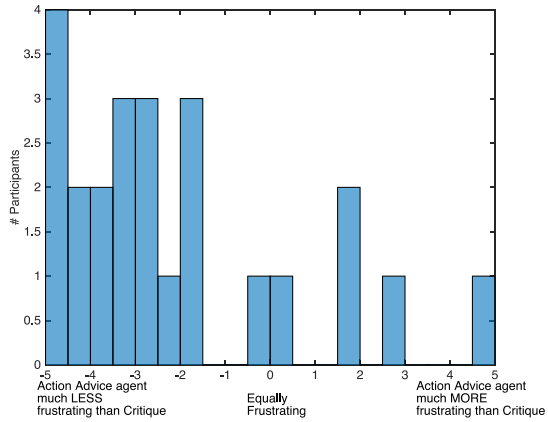

Fig. 6. Frustration direct comparison.

Table 1. Paired T-Tests on Human Experience

Human Factors Metric	Accept/Reject	$t(23)$	p	μ_{advice}	$\mu_{critique}$
Frustration	Reject	−3.1364	0.0046	3.41	5.60
Transparency	Reject	5.4963	1.3738e-05	7.44	3.50
Perceived Intelligence	Reject	4.5527	1.4192e-04	5.48	3.40
Perceived Performance	Reject	2.2401	0.0350	6.85	5.26
Immediacy	Reject	6.2911	2.0291e-06	6.25	3.06

Table 2. One-Sample T-Tests on Human Experience

Human Factors Metric	Accept/Reject	$t(23)$	p	μ
Frustration	Reject	−4.2313	3.1634e-04	−2.25
Transparency	Reject	7.3599	1.7397e-07	2.90
Perceived Intelligence	Reject	5.5087	1.3325e-05	2.387
Perceived Performance	Reject	8.1462	3.1395e-08	3.06
Immediacy	Reject	7.2958	2.0073e-07	2.94

enrolled. We found significant differences between the two agents for every metric. Table 1 shows the results of each t-test.

One-Sample Tests on Human Experience. A series of one-sample t-tests were conducted for the five human factors metrics that were collected following both training sessions. Here the null hypothesis was that the data came from a population with mean equal to zero. A significance of $\alpha = 0.05$ was chosen as a reasonable limit on Type I error given the number of participants enrolled. We found significant differences between the two agents for every metric. Table 2 shows the results of each t-test. These one-sample tests provide a separate and internal verification of the paired tests. The results give us a greater assurance that the differences in the human experience were not influenced by the order in which the participant experienced the agent.

Overall, participants found the action advice agent to create a better human experience than the critique agent. In both the paired and one-sample tests, participants found the action advice agent

to be less frustrating, more immediate, more transparent, and with a higher perceived performance and perceived intelligence compared to the critique agent.

5.2 Frustration

Most participants found action advice to be much less frustrating than critique. The participants' free responses provided valuable insight as to why. We analyzed these responses and found several factors were repeated by many participants. The main factors that made people perceive one agent as more frustrating than another are listed below. We will discuss each in turn.

Powerlessness: Whether the agent's behavior made the human operator feel powerless
Transparency: Whether the human understands why the agent made its choices
Complexity: The complexity of allowed human instruction
Compliance with input: Whether the agent did what it was told
Probabilistic: Whether the agent incorporated input probabilistically
Sensitivity to ASR: The accuracy of the software that transcribes verbal speech to text

5.2.1 Powerlessness. The critique agent made people feel powerless, which caused higher levels of frustration compared to the advice agent. The human's lack of control over the critique agent caused higher levels of frustration compared to the action advice agent.

> **P14** "In the critique case, I felt powerless to direct future actions, especially to avoid the agent jumping into the radioactive pit."
> **P9** "The critique agent did not listen to me, which made me frustrated. It took way longer to respond and did not end up learning how to do the task."
> **P8** "Critique agent was hard to train since there wasn't much feedback as to if my critique had any effect on it at all."
> **P2** "The critique agent was more frustrating to train because I had less direct control and it was not clear how it was interpreting my input."

5.2.2 Transparency. The critique agent caused people to be frustrated because the human teachers could not figure out how their input was being used by the agent.

> **P22** "The Critique agent was very frustrating. I never understood why it made the choices it did, although that could have been because I was not telling it the best commands."
> **P8** "(The) Critique agent was hard to train since there wasn't much feedback as to if my critique had any effect on it at all."
> **P15** "I did not understand how the critique would use my inputs."
> **P10** "Critique was more frustrating because it was very difficult to tell how it was using its knowledge base."
> **P2** "The critique agent was more frustrating to train because I had less direct control and it was not clear how it was interpreting my input."

5.2.3 Complexity. Participants felt that the more complex action advice was less frustrating than the less complex method of critique.

> **P12** "I wanted to give more complex advice to 'help' the Critique Agent."
> **P11** "I felt that the critique agent was more frustrating because I wasn't able to give it specific feedback on how to improve or what it had done correctly."

5.2.4 Compliance with Input. The action advice agent created a less frustrating interaction because it did what it was told, unlike the critique agent.

P11 "The action advice robot just worked. The critique robot took time to train– and even then, he was never as efficient as the action advice robot."
P23 "I never did get the critique agent to the injured person. It just keep going in different directions and I couldn't figure out why."
P9 "The critique agent did not listen to me, which made me frustrated. It took way longer to respond and did not end up learning how to do the task."
P6 "I believe the critique agent took too much time to recognize my input, and did not act upon at all."
P3 "Repeating myself and still getting unintended results."

5.2.5 Probabilistic. Several participants mentioned that the random or probabilistic nature of the critique agent caused them to be more frustrated. The seeming randomness of the agent's response to instruction made them question why the critique agent was choosing certain actions. This is a particularly important lesson for ML researchers because most interaction methods incorporate human instruction in a probabilistic manner (following instructions, choosing between human and RL policies, tapering off human instructions, choosing between human and RL exploration, etc.). While a probabilistic nature of an interactive ML algorithm may be beneficial for testing algorithms in simulation (and is common practice for pure ML), it overlooks a basic and essential aspect of the human interaction experience: if I tell the agent to do something, I want it to follow my instructions in a reliable, repeatable, nonprobabilistic way. We do not suggest that the underlying ML algorithm should be devoid of a probabilistic nature, but rather that the interaction method between the human and agent should be deterministic or heavily biased toward human input in the short term.

P24 "The Critique agent was more frustrating because its development of a mission plan based on user input seemed to be more randomized than truly crafted based on feedback. This made it frustrating to give input, as it felt like it wasn't being used effectively."
P23 "I never did get the critique agent to the injured person. It just keep going in different directions and I couldn't figure out why."

5.2.6 Sensitivity to Speech Recognition. The few participants who found the action advice agent to be more frustrating than critique had problems with the automatic speech recognition software that transcribed their verbal feedback into text. While the ASR software used had a good average accuracy, results vary based on a person's specific accent and speech patterns. In the future, this problem will diminish as ASR models are built using a wider array of training data.

The same ASR software was used for both advice and critique. However, the human's perception of the advice agent is more sensitive to ASR accuracy because it is immediately apparent that the agent did not understand the person if the person says "right" and the agent does not move right. Action advice allows people to provide instructions for the agent's immediate future actions, which makes it sensitive to ASR accuracy because a person can immediately tell if the ASR software was correct. If the ASR software has poor accuracy, it is not as readily apparent for the critique agent, which applies the feedback to past actions.

P21 "Action Advice was far more frustrating because it was not recognizing my commands (especially "Down"). If it HAD been recognizing a very high percentage of my commands, it would have been much, much LESS frustrating than Critique was."
P17 "The action didn't seem to register what I was saying as clearly."

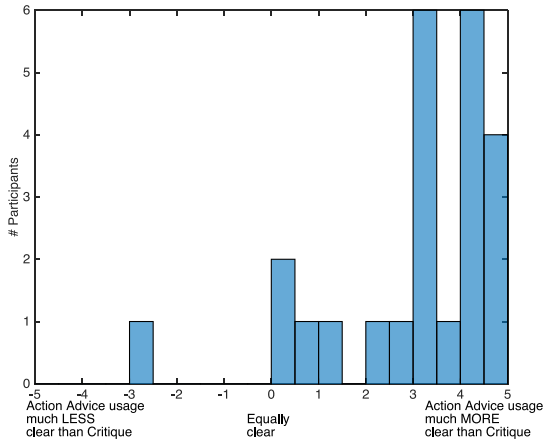

Fig. 7. Transparency direct comparison.

P1 "Action advice was more frustrating because it often couldn't hear me or misheard me."

5.3 Transparency

Every participant except one found that action advice used the human instruction in a clearer manner than critique. The main factor that made people perceive one agent as more clear than another was:

Compliance with input: whether the agent did what it was told

5.3.1 Compliance with Input.

P11 "The action advice agent used my feedback more clearly because it would do exactly what I said. With the critique agent it was hard to tell how it was choosing what to do next."

P24 "The Action Advice agent used feedback much more clearly since it awaited instruction and did exactly what was told, so long as the audio was clear."

P23 "I could see that the Action Agent was at least responding by going the way I said. Even if I told the critique agent it was bad, it didn't necessarily try something different."

P21 "Action Advice was using my feedback, when it understood it correctly, to directly move. I had to guess what Critique was going to do with my input."

P20 "I could see action agent respond to my input, but the critique agent didn't seem to use the input."

P15 "(With the action advice agent,) a certain input caused an output. The critique agent seemed to be going randomly with small help."

P9 "It seems that the action agent used my feedback better because I was able to get it to consistently collect the object and make it to the goal. The critique agent seemed to do what it wanted and did not take my advice."

P5 "The (action advice) agent responded in the right direction with a quicker time."

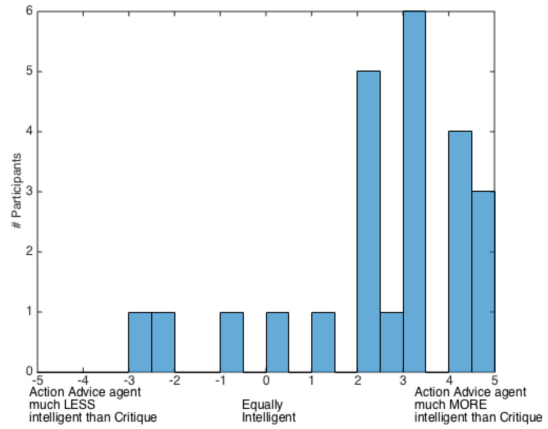

Fig. 8. Perceived intelligence direct comparison.

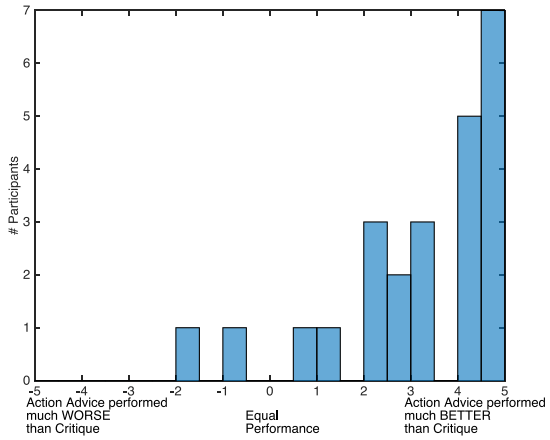

Fig. 9. Perceived performance direct comparison.

P4 "The speed and accuracy of the (advice) agent's movements in response to my feedback. Essentially, seeing the agent do what I 'wanted it to do' made it seem like my feedback was used more clearly."

P2 "The action advice agent tended to move in the direction I had previously advised it to move, but the critique agent still moved in general directions that I had given negative feedback for previously."

5.4 Perceived Intelligence

The majority of participants perceived the advice agent to be more intelligent than the critique agent, even though both used the same underlying ML algorithm. Without human input, both agents were equally capable, but the teaching method caused the human-agent interaction to differ. Most participants found the action advice agent to be more intelligent than the critique agent, with 54% giving scores +3 or greater. Only three participants rated the critique agent as more intelligent, and none of these rated it strongly so. The main factors that made people perceive one agent as more intelligent are listed below.

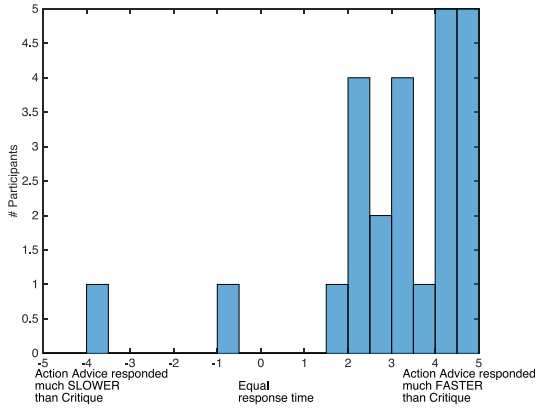

Fig. 10. Immediacy direct comparison.

Compliance with input: Whether the agent did what it was told
Immediacy: How quickly the agent learned
Effort: The amount of input needed to train the agent
Complexity: The complexity of allowed human instruction
Transparency: Whether the human understands why the agent made its choices
Robustness and Flexibility: The agent's ability to correct mistakes and learn alternate policies

A detailed analysis of the perceived intelligence results can be found in our associated article (Krening and Feigh 2018).

5.5 Perceived Performance

Most participants thought action advice performed better than critique. In fact, no participants found the action advice agent to perform much worse than critique, and over 60% of participants felt action advice performed much better than critique. We did not ask participants for free-form responses on perceived performance, assuming the responses to "If one agent performed the task as you intended better than the other, what made you think that?" would be redundant.

5.6 Immediacy

Only two participants found action advice to respond slower than critique, and they had issues with the ASR software accuracy. Similar to perceived performance, we did not ask participants for free responses about immediacy, assuming tautological answers to "If one agent responded to your input faster than another, what made you think that?" However, through the participants' free responses to other human factors metrics, we have found the immediacy of the agent's response to be a main factor impacting both frustration and perceived intelligence.

5.7 Objective Metrics on ML Performance

Paired Tests on Interval ML Data: Figure 11 illustrates the results of the objective metrics of each participant training each agent. Paired t-tests were conducted for the ML metrics that were interval in nature: training time and average reward. The null hypothesis was that the pairwise difference between the two paired groups had a mean equal to zero. We found significant differences of the training time and reward between the two agents. Table 3 shows the results of each t-test.

Paired Tests on Ordinal ML Data: Wilcoxon Signed Rank tests were conducted for the objective metrics that were ordinal in nature: average number of training inputs from the human and average

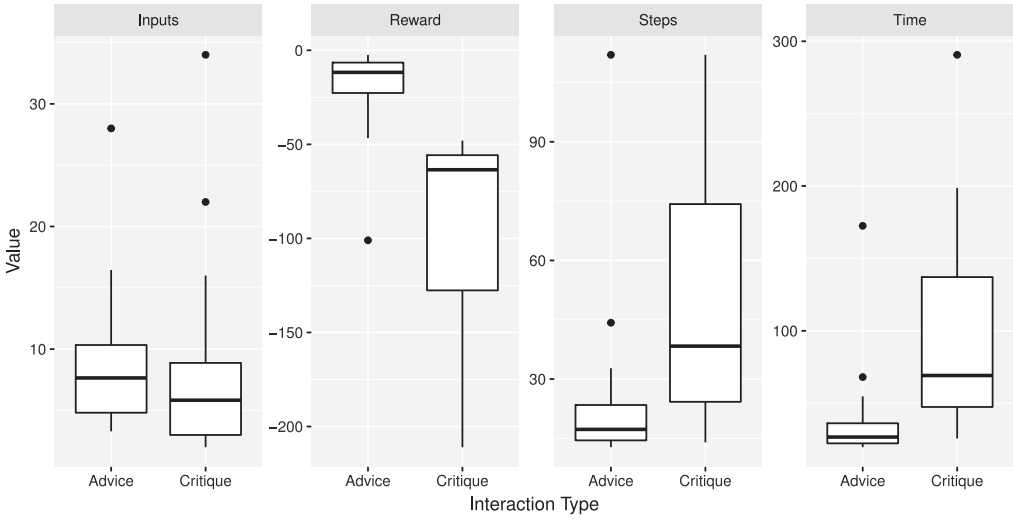

Fig. 11. ML performance comparison.

Table 3. Paired T-Tests on ML Performance

Objective Metrics	Accept/Reject	$t(23)$	p	μ_{advice}	$\mu_{critique}$
Training Time (s)	Reject	−3.9321	7.6406e-04	38.51	99.81
Avg Reward	Reject	7.3106	3.3849e-07	−18.92	−89.57

Table 4. Wilcoxon Signed Rank Tests on ML Performance

Objective Metrics	Accept/Reject	Z	p
Avg Number Inputs	Accept	1.1201	0.2627
Avg Number Steps	Reject	−2.9057	0.0037

number of steps to complete each episode. The null hypothesis was that the difference between the pairwise samples came from a distribution with zero median. We found that there was a significant difference between advice and critique in the number of steps it took to complete a level. However, there was not a significant difference in the amount of input given by the human teacher. Table 4 shows the results of each test.

5.7.1 Training Time. The training time was significantly shorter for advice than critique.

5.7.2 Amount of Human Input. The amount of input provided by the human teacher was approximately the same for advice and critique.

5.7.3 Number of Steps to Complete an Episode. The advice agent was able to complete an episode in significantly fewer steps than critique. The critique agent spends many steps wandering around the domain, seemingly aimlessly. The median number of steps for advice was 24, while the critique agent doubled that with a median of 50.

5.7.4 Reward. The advice agent received a significantly higher reward than critique.

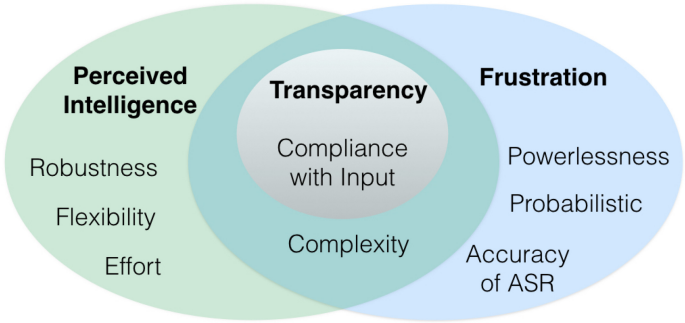

Fig. 12. Summary of factors that impact frustration, transparency, and perceived intelligence based on participants' free responses.

6 DISCUSSION

6.1 Summary of Participants' Long Responses

The nested Venn diagram in Figure 12 shows a summary of which factors impact the human teacher's frustration as well as perceptions of transparency and intelligence based on the free responses. The main factor impacting transparency was compliance with input, and these two factors along with complexity impacted both the human's frustration and perceived intelligence of the agent. Feelings of powerlessness, a probabilistic use of the user's instruction, and a sensitivity to the ASR software increased frustration. Robustness, flexibility, and effort impacted perceived intelligence.

We find it interesting that complexity does not detract from the human experience. The more complex form of input, action advice, created a better user experience by lowering frustration and creating a perception of a more intelligent agent. There is likely a point at which the input could be designed to become so complex and detailed that it becomes unwieldy and detracts from the user experience. On the other side of the complexity scale, the results show that input of a too-simplistic form can damage the human's satisfaction with the agent.

6.2 Design Considerations

The results of this study indicate that specific characteristics of an interaction algorithm will impact the human teacher's experience. Before researchers or designers create, modify, or choose an interaction algorithm, we suggest they consider the following characteristics to improve the human's experience of working with the agent:

(1) **Instructions about future, not past.** An algorithm that uses instructions about the future (such as action advice) instead of the past fosters a positive human experience because it increases the perceived control the human has over the agent, allows for greater transparency of how the agent uses the instructions, and enables the human to immediately detect issues with the agent's compliance. Choosing a rhetorically positive teaching method will also set the tone for a more positive human-agent interaction.
(2) **Compliance with input.** If the human teacher sees the agent following instructions, the person will be less frustrated and perceive the agent to have a higher performance and intelligence.
(3) **Empowerment.** An interaction algorithm that forces the agent to clearly, immediately, and consistently follow the human's instructions will decrease feelings of powerlessness, which will in turn decrease the person's frustration.

(4) **Transparency.** An algorithm in which the human teacher can clearly understand how the agent used the human's instructions to choose an action will decrease frustration and increase perceived intelligence. Greater transparency can be achieved if an agent immediately complies with the instructions.
(5) **Immediacy.** The human's experience will be improved if the agent immediately responds to the instructions because it creates a sort of instant gratification for the human.
(6) **Deterministic interaction.** While the underlying ML algorithm will doubtlessly be probabilistic, the interaction with the human should be such that the agent follows the instructions in a reliable, repeatable, nonprobabilistic manner. A deterministic interaction with the human will decrease frustration and decrease the person's uncertainty in the agent that degrades trust.
(7) **Complexity.** An algorithm that allows for more complex instructions than binary good versus bad critique will decrease frustration and increase perceived intelligence.
(8) **ASR accuracy.** When choosing ASR software, it is worth the effort to improve the accuracy in order to decrease the human's frustration. Also, while a person is more aware of the accuracy and processing duration for an advice agent, this does not mean the critique agent is better in this regard. Rather, the critique agent trades processing lag for less transparent agent response, which increases frustration and decreases perceived intelligence.
(9) **Robustness and flexibility.** The ability to correct mistakes and teach the agent alternate policies improves the human teacher's experience.

6.3 Thoughts on Using Critique for IML Applications

At first glance and from an ML researcher's perspective, critique is a very attractive option for using human feedback. It is simpler, binary feedback (positive vs. negative). It does not require grounding feedback to the state/action space, and would not need to be altered when switching between domains or to embodied robots. Critique can be incorporated into machine-learning algorithms in several ways; it can be used directly as a reward signal or, more efficiently, as policy information in a Policy Shaping algorithm. However, Policy Shaping does not promote a positive human experience. We need a better way to connect critique to ML. Is it worth the time and effort to pursue better critique algorithms? Critique is perceived as less intelligent than action advice, and is not as human-friendly. People cannot directly control a critique agent, leading to feelings of powerlessness. It is unclear how critique is being used by the agent during training, and people become frustrated with the lack of transparency. People give instructions about the future, even if expressly told not to Thomaz and Breazeal (2008), so it is not intuitive to use an interaction method that disallows future instructions. There is no definitive way to solve the credit attribution problem. Also, critique is inherently negative from a rhetorical perspective. It is tempting to focus on making action advice better instead of pursuing critique. In the end, it would be beneficial to use both advice and critique since people naturally switch between both, but that is beyond the scope of this work.

6.4 Limitations

We have shown that the interaction algorithm impacts the human's satisfaction with the agent. Specifically, the Newtonian Action Advice agent creates a better experience for the human than Policy Shaping. In terms of teaching methods, our experiment used Newtonian Action Advice as an ambassador of action advice, which can be thought of as verbal demonstrations, and Policy Shaping as a representative of critique. We cannot use our results to directly guarantee that all action advice algorithms will create a better user experience than all critique algorithms. However, after analyzing the participants' explanations of what impacted the user experience, we identified

nine design considerations that each indicate action advice is the superior teaching method and is inherently more likely to create a better experience for the human teacher than critique.

The experimental scenario was chosen to even the playing field between the advice and critique agents. A domain with a limited dimension was chosen so a critique agent could be trained in less than 15 minutes. Four primitive actions were used so the advice agent was limited to four possible inputs compared to critique's two.

A limitation of the action advice agent is that a menu of available actions must be defined by the researcher. The available natural language descriptions of the actions were grounded to the agent's corresponding primitive actions prior to the experiment. For example, the word " right" was grounded to the agent's action that would move the agent one square to the right. The critique agent did not share this limitation.

7 CONCLUSION

We have shown that the interaction algorithm can impact the human's experience with an IML agent. The Newtonian Action Advice agent created a better experience for the human than the critique-driven Policy Shaping.

We have identified nine main characteristics that impact the human's experience with the agent, including using human instructions about the future, compliance with input, empowerment, transparency, immediacy, a deterministic interaction, the complexity of the instructions, accuracy of the speech recognition software, and the robust and flexible nature of the interaction algorithm.

These design characteristics suggest that, in terms of teaching methods, action advice is likely to create a better experience for the human teacher than critique.

This article demonstrates that it is not enough to design algorithms that can theoretically use human input; we must go further and design algorithms that create a positive human experience. IML algorithms must be verified using human factors metrics such as frustration in addition to traditional ML metrics such as cumulative reward.

ACKNOWLEDGMENTS

This work was funded under ONR grant number N000141410003.

REFERENCES

Brenna D. Argall, Brett Browning, and Manuela Veloso. 2008. Learning robot motion control with demonstration and advice-operators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)*. IEEE, 399–404.

Christoph Bartneck, Takayuki Kanda, Omar Mubin, and Abdullah Al Mahmud. 2009. Does the design of a robot influence its animacy and perceived intelligence? *International Journal of Social Robotics* 1, 2 (2009), 195–204.

Thomas Cederborg, Ishaan Grover, Charles L. Isbell, and Andrea Lockerd Thomaz. 2015. Policy shaping with human teachers. In *International Joint Conferences on Artificial Intelligence (IJCAI'15)*. 3366–3372.

Richard Dearden, Nir Friedman, and Stuart Russell. 1998. Bayesian Q-learning. In *AAAI/IAAI*. 761–768.

Carl F. DiSalvo, Francine Gemperle, Jodi Forlizzi, and Sara Kiesler. 2002. All robots are not created equal: The design and perception of humanoid robot heads. In *Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. ACM, 321–326.

Brian R. Duffy. 2003. Anthropomorphism and the social robot. *Robotics and Autonomous Systems* 42, 3 (2003), 177–190.

Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. 2003. A survey of socially interactive robots. *Robotics and Autonomous Systems* 42, 3 (2003), 143–166.

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

Sandra G. Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 50. Sage Publications, Los Angeles, CA, 904–908.

Jay Heinrichs. 2017. *Thank You for Arguing: What Aristotle, Lincoln, and Homer Simpson Can Teach Us about the Art of Persuasion*. Three Rivers Press (CA).

David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W. Black, Mosur Ravishankar, and Alexander I. Rudnicky. 2006. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)*. Vol. 1. IEEE, I–I.

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.

Charles Lee Isbell, Michael Kearns, Satinder Singh, Christian R. Shelton, Peter Stone, and Dave Kormann. 2006. Cobot in LambdaMOO: An adaptive social statistics agent. *Autonomous Agents and Multi-Agent Systems* 13, 3 (2006), 327–354.

Madhura Joshi, Rakesh Khobragade, Saurabh Sarda, Umesh Deshpande, and Swati Mohan. 2012. Object-oriented representation and hierarchical reinforcement learning in Infinite Mario. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI'12)*. Vol. 1. IEEE, 1076–1081.

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*. Vol. 1. International Foundation for Autonomous Agents and Multiagent Systems, 5–12.

Samantha Krening. 2018. Newtonian action advice: Integrating human verbal instruction with reinforcement learning. arXiv, arXiv:1804.05821.

Samantha Krening and Karen M. Feigh. 2018. Characteristics that influence perceived intelligence in AI design. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*.

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 Transactions on Cognitive and Developmental Systems* 9, 1 (2017), 44–55.

Gregory Kuhlmann, Peter Stone, Raymond Mooney, and Jude Shavlik. 2004. Guiding a reinforcement learner with natural language advice: Initial results in RoboCup soccer. In *The AAAI 2004 Workshop on Supervisory Control of Learning and Adaptive Systems*.

James MacGlashan, Monica Babes-Vroman, Marie desJardins, Michael Littman, Smaranda Muresan, Shawn Squire, Stefanie Tellex, Dilip Arumugam, and Lei Yang. 2015. Grounding English commands to reward functions. In *Proceedings of Robotics: Science and Systems*.

Richard Maclin, Jude Shavlik, Lisa Torrey, Trevor Walker, and Edward Wild. 2005. Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *Association for the Advancement of Artificial Intelligence (AAAI'05)*. 819–824.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 55–60. Retrieved from http://www.aclweb.org/anthology/P/P14/P14-5010.

Cetin Meriçli, Steven D. Klee, Jack Paparian, and Manuela Veloso. 2014. An interactive approach for situated task specification through verbal instructions. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1069–1076.

Andrew Y. Ng and Stuart J. Russell. 2000. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning (ICML'00)*. 663–670.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 115–124.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2, 1–2 (2008), 1–135.

Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors* 39, 2 (1997), 230–253.

Taylor Phillips. 2010. Put your money where your mouth is: The effects of southern vs. standard accent on perceptions of speakers. *Social Sciences* (2010), 53–56. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C6&q=Taylor+Phillips.+2010.+Put+your+money+where+your+mouth+is%3A+The+effects+of+southern+vs.+standard+accent+on+perceptions+of+speakers.+S&btnG=

Tamara Rakić, Melanie C. Steffens, and Amélie Mummendey. 2011. Blinded by the accent! The minor role of looks in ethnic categorization. *Journal of Personality and Social Psychology* 100, 1 (2011), 16.

Himanshu Sahni, Brent Harrison, Kaushik Subramanian, Thomas Cederborg, Charles Isbell, and Andrea Thomaz. 2016. Policy shaping in domains with multiple optimal policies. In *Proceedings of the 2016 International Conference on AAMAS*. International Foundation for AAMAS, 1455–1456.

Manimaran Sivasamy Sivamurugan and Balaraman Ravindran. 2012. Instructing a reinforcement learner. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference (FLAIRS'12)*.

Burrhus Frederic Skinner. 1990. The behavior of organisms: An experimental analysis. BF Skinner Foundation.

Burrhus F. Skinner. 1981. Selection by consequences. *Science* 213, 4507 (1981), 501–504.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*. Vol. 1631. Citeseer, 1642.

Kaushik Subramanian, Charles L. Isbell Jr., and Andrea L. Thomaz. 2016. Exploration from demonstration for interactive reinforcement learning. In *Proceedings of the 2016 International Conference on AAMAS*. International Foundation for AAMAS, 447–456.

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

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth J. Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Association for the Advancement of Artificial Intelligence (AAAI'11)*. Vol. 1. 2.

Jesse Thomason, Jivko Sinapov, Maxwell Svetlik, Peter Stone, and Raymond J. Mooney. 2016. Learning multi-modal grounded linguistic semantics by playing "I Spy." In *International Joint Conferences on Artificial Intelligence (IJCAI'16)*. 3477–3483.

Jesse Thomason, Shiqi Zhang, Raymond J. Mooney, and Peter Stone. 2015. Learning to interpret natural language commands through human-robot dialog. In *International Joint Conferences on Artificial Intelligence (IJCAI'15)*. 1923–1929.

Andrea L. Thomaz and Cynthia Breazeal. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence* 172, 6–7 (2008), 716–737.

Received April 2018; revised August 2018; accepted August 2018