Effect of Interaction Design of Reinforcement Learning Agents on Human Satisfaction in Partially Observable Domains

Divya Srivastava, Spencer Frazier, Mark Riedl and Karen M. Feigh
Georgia Institute of Technology, Atlanta, Georgia, U.S.A.

Keywords: Human-robot Interaction, Interactive Reinforcement Learning, Interactive Machine Learning, Algorithm Design, User Experience.

Abstract: Interactive machine learning involves humans teaching with agents during their learning process. As this field grows, it is pertinent that laymen teachers, i.e. those without programming or extensive ML experience, are able to easily and effectively teach the agents. Previous work has investigated which factors contribute to the teacher's experience when training agents in a fully observable domain. In this paper, we investigate how four different interaction methods affect agent performance and teacher experience in partially observable domains. As the domain in which the agent is learning becomes more complex, it accumulates less reward overall and needs more advice from the teacher. It is found that the most salient features that affect teacher satisfaction are agent compliance to advice, response speed, instruction quantity required, and reliability in agent response. It is suggested that machine learning algorithms incorporate a short time delay in the agent's response and maximize the agent's adherence to advice to increase reliability of the agent's behavior. The need to generalize advice over time to reduce the amount of instruction needed varies depending on the presence of penalties in the environment.

1 INTRODUCTION

Traditionally, the role of a human in machine learning (ML) is limited to loading data, picking which features to extract, and post-processing the output of the ML algorithm. There is no human interaction during the actual learning process. ML experts rely on their contextual knowledge of the problem to evaluate the results and then reiterate on the ML algorithm until the learning process converges on expected behavior. In interactive machine learning (IML), humans play an active role during the learning process of machine learning agents (Thomaz et al., 2005). Interactive reinforcement learning (IRL) is a subset of IML that has proven to converge quicker on desired behavior than traditional ML techniques with no human interaction (Krening and Feigh, 2018), which indicates value in having users interact with agents during their learning process. The overall field of IML focuses on human-in-the-loop studies because as artificial intelligence becomes more commonplace in everyday life, we want non-experts to be able to interact with and train agents easily.

Typically, reinforcement learning (RL) algorithms are evaluated using objective RL metrics, such as cumulative reward, total steps per episode, and training time. When previous studies have sought to evaluate IML agents, they have relied heavily on oracles to aid agents during the learning process, and have shown significant training efficiency impacts when the frequency and accuracy of the advice are varied (Frazier and Riedl, 2019). But while the consistency and repeatability of oracles may be useful to understand the effects of possible human behavior on IML agents, studies have shown that this is not equivalent to humans directly interacting with the agents (Amershi et al., 2014). Additionally, simulated human input cannot answer questions regarding the experience of a human teacher, who can have emotional responses such as frustration or confusion. Human experiences teaching agents have been shown to vary based on the training modality or interaction mechanism used (Krening and Feigh, 2019).

An added challenge of human-in-the-loop training is that humans adapt their teaching style based on the context of the task domain and their assessment of the capabilities of the agent (Thomaz and Breazeal, 2006). For example, if a student is doing a task well on their own without a teacher's help, the teacher may leave them to their own devices and just monitor the student in case they do end up needing help. Conversely, if a student is struggling with a task or is

Srivastava, D., Frazier, S., Riedl, M. and Feigh, K.
Effect of Interaction Design of Reinforcement Learning Agents on Human Satisfaction in Partially Observable Domains.
DOI: 10.5220/0010240101740181
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 2: HUCAPP*, pages 174-181
ISBN: 978-989-758-488-6

headed in the wrong direction, a teacher tends to step in and guide them with more explicit and frequent instruction to help them accomplish the task. The teacher's teaching style changes depending on what is needed from them at the time.

Ideally, when teaching an agent to do a task, the teacher would have all the necessary and relevant information needed to complete the task. They would know what to do with the information at hand to accomplish the goal. Domains with this property are known as fully observable domains. However, realistically, many domains do not have such traits. Humans are often presented with situations in which they have only partial information of the environment, known as a partially observable domain. They are required to make decisions to satisfy their goals, which is often done by filling in knowledge gaps as they proceed with their task (Klein and O'Brien, 2018). When teaching agents, we want to mimic real world situations in which the human teacher has limited knowledge and must give advice to the agent based on their current understanding of the environment, as more and more information about the environment is revealed. Real world environments also yield constraints in the form of direct consequences for actions.

Previous work has investigated how different methods of an interaction algorithm affect human experience in teaching ML agents in a fully observable environment with penalties (Krening and Feigh, 2019). The penalties are a result of hazards in the environment that simulate real world constraints. It was found that the human's level of frustration teaching the agent, and how intelligent the teacher thinks the agent is, are affected greatly by the method used to interact with the agent. Specifically, in a fully observable domain with penalties, it was found that quick agent response times and the adherence to the advice given were highly correlated with human teaching satisfaction.

This paper expands upon (Krening and Feigh, 2019) to investigate which interaction features affect user experience in teaching ML agents in partially observable domains with and without penalties. It is hypothesized that the findings of the fully-observable domain will not hold in a partially observable domain, and that the introduction of penalties will cause the human teacher to be more conservative in their advice.

2 METHOD

In this study, we conducted two repeated measures, within-subject experiments in which we investigated the effect of 4 different interaction methods on the participant's experience of teaching the agent. The experiment took place in-person, and collected data from 24 participants and 30 participants for each study, respectively. All participants were ML novices, and the ordering of the trials was randomized according to a Latin square design. We made a concerted effort to recruit individuals from the general population.

The participants were required to teach each agent to navigate a maze developed in the Malmo minecraft platform. In the game, there are 2 players: the agent, and a non-playable character. The agent needs to navigate through the maze to find and approach the non-playable character. In the first experiment, there were no penalties. In the second, there were penalties associated with a water hazard. If the agent entered the water, the agent fails the task is penalized with negative reward. The participant is able to see the maze in two ways: an isometric view, in which part of the maze is obscured, and the agent's point of view from within the maze (Figure 1) – both of which provide only partial observability. The goal of the task is to find and approach a non-playable character situated near the end of the maze.

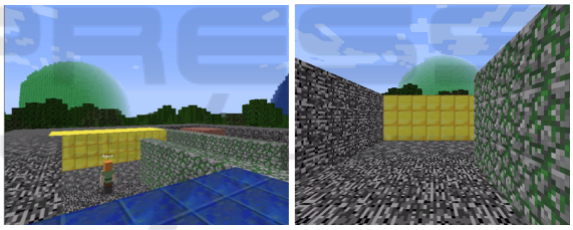

Figure 1: Partially Observable Maze in Minecraft. Left window shows isometric view. Right window shows agent's POV from within the maze.

The teachers provided advice to the agent in the form of arrow key presses on the keyboard, which were sent to an interaction algorithm (for all of the algorithm methods). The interaction algorithm collaborated with the reinforcement learning agent to select which action to take. All agents share the same action-selection process (see 2.2), and so all are equally capable of completing the task.

Participants were told to repeat the task for as many training episodes as they felt was necessary to achieve satisfactory performance from the agent. They were also told that they could stop training if they were too frustrated to continue, or for any other reason. As a result, the training time per episode varied for every participant and interaction algorithm. After each agent was trained, the participant was asked to complete a questionnaire about their experience. At the end of the experiment, participants were

given a final questionnaire comparing all four interaction methods.

2.1 Action Advice Interaction Methods

In this work, we implement four interaction methods: 5-Step, 1-Step, Probabilistic, and Time Delay (Krening and Feigh, 2019), summarized in Table 1 . In the 5-Step method, the agent moves for 5 steps in the direction advised by the teacher. For example, if the teacher advises the agent to "move forward" (up arrow), the agent will do so for 5 steps before reverting back to its action selection process. The 5 steps may be interrupted by the teacher at any time with new advice. In the 1-Step method, the agent moves for 1 step in the direction advised by the teacher before returning back to its action selection process. In the Probabilistic method, the agent has a 60% chance of following the teacher's advice. The Probabilistic variation is implemented to mimic the stochastic nature of an agent in a real-world environment, and tests how inconsistent agent behavior affects teachers. In the Time Delay method, a two-second delay is introduced between the time the advice is given and the time the agent executes that advice. During that delay, the agent stands still in its current position.

Table 1: Four Variations of Intearction Algorithm.

1-Step: Advice is followed for 1 step in the direction advised by the teacher	5-Step: Advice is followed for 5 steps in the direction advised by the teacher
Time Delay: Advice is followed for 1 step in the direction advised by the teacher after a 2-second delay	Probabilistic: Agent chooses whether to follow advice based on a probability

2.2 Action Selection Process

Two action-advice augmented reinforcement learning algorithms are used for action selection. The first is the *Newtonian Action Advice* (NAA) algorithm (Krening, 2018) which is used to implement the 5-Step interaction algorithm. The second is based on the *Feedback Arbitration* (FA) algorithm (Lin et al., 2017), a specific form of Q-learning that leverages Deep Neural Networks. In this work, in addition to the standard reinforcement learning training loop, an *arbiter*, continually assesses feedback consistency and quality versus the confidence it has in its learned policy. Feedback Arbitration is already designed to work in 3D virtual environments similar to Minecraft though on much simpler maps.

The Feedback Arbitration (FA) algorithm is a DQN algorithm with an off-policy arbiter. Action advice given by the teacher is queued up in a pending advice array. Queued advice instances dequeue after some time. The agent either consults the pending advice array, explores the environment using a random action, or exploits its Q-network by picking the action it believes has the highest utility in the current state. Random actions are chosen according to the standard ε-greedy exploration vs exploitation strategy, with ε-decaying over time. If the confidence is low, as it is in this work, the agent chooses to consult the action advice provided by human participants. Otherwise, it measures its confidence in its Q-network. Then the Q-network confidence score is computed as:

$$relativeCost = \frac{-1}{\ln \sqrt{(\frac{\min_{a \in A(s)} L_a}{L_{\max}})} - 1} \quad (1)$$

where L_a is the loss value for the predicted activation of action a in the current state and $L_{\max}$ is the highest loss observed by the DQN thus far.

It is worth mentioning that potential-based reward-shaping (Ng et al., 1999) would greatly improve the training efficiency of this agent but was not implemented to adhere to this work's implementation.

2.3 Objective ML Measures

While participants were training the agents, objective performance metrics were logged to data files: 1) training time per episode, 2) cumulative reward earned per episode, 3) the number of times the teacher provided advice per episode, 4) the number of steps the agent took to complete an episode. This data is normalized across both studies to allow for comparison.

2.4 Human Experience Measures

After training each agent, participants completed a questionnaire. The participants were asked to rate the intelligence of the agent on a interval scale from [0:10]. A rating of 0 indicated that the agent was not intelligent, while a rating of 10 meant very intelligent. The same scale of [0:10] was used for four additional metrics: the agent's overall performance, transparency in how it used the teacher's advice, immediacy in responding to the teacher's advice, and the teacher's level of frustration. A score of 0 corresponded to poor performance from the agent, unclear use of feedback, a slower response time, and low frustration on the teacher's part. Scores of 10 indicated excellent performance from the agent, clear use

of feedback, an immediate response time, and high frustration levels for the teacher.

At the end of the experiment, participants were asked to rank all four agents from most to least intelligent, and the easiest to hardest to train. Additionally, participants were asked in a free-response question why they ranked the agents the way they did. This was done to elicit the factors that affected their perception of the agents, and thus, each interaction method. These written responses were entirely free form with no priming by the experimenter.

3 RESULTS AND DISCUSSION

This section presents the results obtained from a previous study (Krening and Feigh, 2019), combined with the two studies conducted in this work. We compare the results to show the change in metrics from a fully observable domain with penalties (Study 0), to a partially observable domain without penalties (Study 1), to a partially observable domain with penalties (Study 2). All data has been normalized using min-max normalization. Tables 2, 3, 4, and 5 show the results of a two-way within subjects ANOVA for Study 1 and 2, respectively across the dependent variables of Number of Steps Taken, Quantity of Advice Given, the Cumulative reward, and Training Time. Unless otherwise noted, all assumptions for this analysis are met and the order of the presentations of the agents was not significant.

3.1 Objective Metrics

In this section of results, all figures include significance bars from post-hoc analyses. One star (*) indicates a P-value $<= 0.05$. Two stars (**) indicates a P-value $<= 0.01$. Three stars (***) indicates a P-values $<= 0.001$. Four stars (****) indicates a P-value $<= 0.0001$.

Table 2: Repeated-Measures ANOVA for Steps Taken.

Study		*SS*	*df*	*F*	*P*
1	Alg.	36321	3	57.5	<2e-16
	Order	52	1	0.25	0.621
2	Alg.	11724	3	33.3	<2e-16
	Order	81	1	0.67	0.413

Table 2 shows the results for the number of steps taken by the agent per episode. It can be seen that the average number of steps the agent took to complete each level is not equal across the different variations of the interaction algorithm. Figure 2 shows

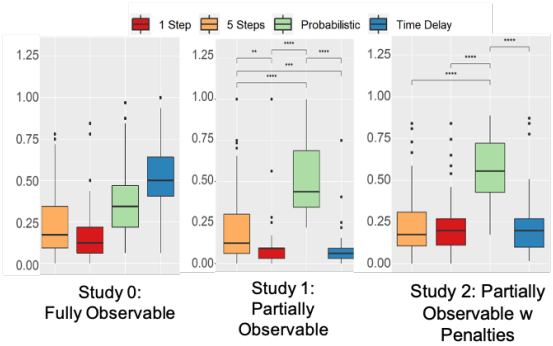

Figure 2: Steps taken to complete an episode in three different domains.

that, as the domain becomes more complex, the number of steps it takes to complete an episode is roughly the same for the 1-Step and Time Delay variation. It can be seen that introducing probability to the user interaction results in an increasing number of steps per episode. Further, introducing a time delay proves most problematic when penalties exist in the environment.

Table 3: Repeated-Measures ANOVA for Advice Given.

Study		*SS*	*df*	*F*	*P*
1	Alg.	4118	3	5.91	6.48e-04
	Order	2147	1	0.25	2.61e-03
2	Alg.	3563	3	4.76	2.34e-03
	Order	778	1	3.12	0.078

Table 3 shows the results for the quantity of advice given by the participants. It can be seen that the average amount of advice the agent took to complete each level is not equal across all of the interaction methods. All methods aside from Probabilistic received roughly the same amount of advice from the teacher.

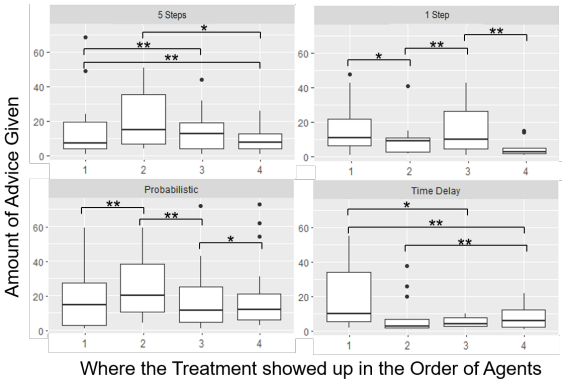

Figure 3: How much advice was given to an agent depending on when they were shown to the teacher.

The order in which the agents were shown to the

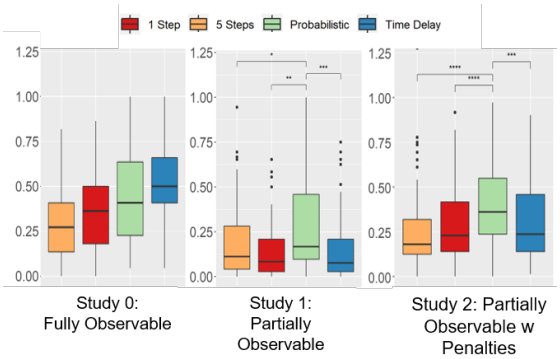

Figure 4: Amount of advice given per episode in three different domains.

participant was significant in Study 1. Further analysis (Figure 3) into this shows that, aside from the Probabilistic treatment, there was a decrease in the amount of advice given to the agents that the participant saw in the latter half of the experiment. This indicates that the participant either became fatigued and/or disengaged in teaching the agents.

Figure 4 shows that, as the domain becomes more complex, the quantity of advice given by the teacher will vary depending on the domain. In the partially observable domain without penalties, people provided less advice for the Time Delay interaction than the other methods, which is surprising because in the fully observable domain, the Time Delay agent was given the most advice. In the partially observable domain, the 1-Step agent was a close second in terms of least amount of advice needed, followed by the 5-Step agent. This is interesting because theoretically, with the 5-Step interaction, the teacher would not be giving advice at each time step. It is possible that people may have given advice in the middle of the agent taking 5 steps in order to change the agent's direction, which resulted in more advice being given. In the partially observable domain with penalties, more advice is needed from the teacher in navigating around the penalties. It is interesting to see that the 5-Step method required the least amount of advice in the domains with penalties, but not in the partially observable domain without penalties.

Table 4: Repeated-Measures ANOVA for Earned Reward.

Study		*SS*	*df*	*F*	*P*
1	Alg.	53987	3	26.2	<8.4e-15
	Order	949	1	1.38	0.241
2	Alg.	14767	3	6.39	3.05e-04
	Order	615	1	0.79	0.372

Table 4 shows the results for the amount of earned

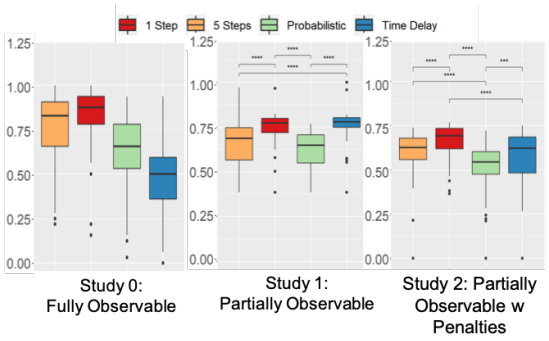

Figure 5: Reward earned per episode in three different domains.

reward. It can be seen that the reward earned per episode is not equal across the variations of the interaction algorithm. A post-hoc pairwise analysis for both Study 1 and 2 shows that there was a significant difference in the reward earned by the probability variation and the reward earned by all the other variations. Figure 5 shows that, as the domain becomes more complex, the cumulative reward decreases across all agents. Furthermore, the 1-Step method continuously accumulates the highest reward.

Table 5: Repeated-Measures ANOVA for Total Training Time.

Study		*SS*	*df*	*F*	*P*
1	Alg.	3.6e6	3	195.7	<2e-16
	Order	7916	1	1.294	0.256
2	Alg.	9.8e4	3	21.46	5.92e-13
	Order	23	1	0.015	0.903

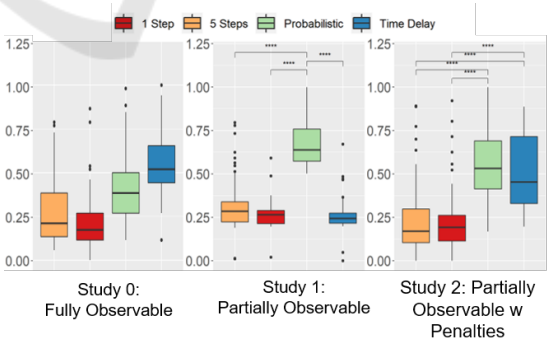

Figure 6: Training time per episode in three different domains.

Table 5 shows the results for total training time. It can be seen that the training time per episode is not equal across all of the interaction methods. Figure 6 shows that, in a partially observable domain, the Time Delay agent took the least amount of time to train per episode. This may be because the amount of advice

given to the Time Delay agent was consistently lower than the amount given to the 1-Step agent (Figure 3). The advice also dropped off significantly and thus the agent fell back on its FA algorithm and equated the 1-Step agent in training time. However, the Time Delay agent took much longer to train in Study 2. The 1-Step method consistently had the best training time, followed by the 5-Step method.

It should be noted that, in the domain without penalties, the Time Delay method resulted in roughly the same agent performance as the 1-Step method across all the aforementioned dependant variables. However, in domains with penalties, the Time Delay agent performs either slightly or significantly worse than the 1-Step, depending on observability. From a performance standpoint, the 1-Step is objectively better since it performs well in all domains. However, as will be shown in the next section, introducing a short time delay can make for an overall better experience for the teacher.

3.2 Human Experience Metrics

After teaching each agent, participants answered a questionnaire about their experience with that agent. Tables 6-10 show the results of a one-way within subjects ANOVA for Study 1 and 2, respectively across the dependent variables of the user's frustration level, and perceived intelligence, transparency, response time, and perceived performance of the agent. From the ANOVA results, it is apparent that the design of the interaction algorithm was significant to each human measure. Figure 7 shows the composite of participants' answers to the questionnaire. For all human factors metrics except frustration, higher values indicate a better human experience (better perceived performance, perceived agent intelligence, and understanding of agent). For frustration, higher values indicate a higher level of frustration, and therefore a worse human experience.

Table 6: Repeated-Measures ANOVA for Perceived Intelligence.

Study		*SS*	*df*	*F*	*P*
1	Alg.	404	3	26.66	7.20e-13
2	Alg.	270	3	17.97	1.20e-09

The Probabilistic method consistently rated worse than other interaction methods in partially observable domains. Unreliable behavior from the agent kept the teacher engaged for the duration of the task (Figure 3), which indicates that the teacher was able to pick up on the lack of predictability in the agent's behavior, but this led to high frustration. Frustration levels were much higher compared to the other methods, and the agent received low scores in terms of every other dimension measured. This indicates that unreliable behavior from an agent makes for a poorer teaching experience, especially in a partially observable domain.

Table 7: Repeated-Measures ANOVA for Frustration.

Study		*SS*	*df*	*F*	*P*
1	Alg.	436	3	27.62	3.22e-13
2	Alg.	216	3	9.341	1.40e-05

Table 8: Repeated-Measures ANOVA for Transparency.

Study		*SS*	*df*	*F*	*P*
1	Alg.	369	3	22.33	3.07e-11
2	Alg.	216	3	9.32	1.43e-05

Table 9: Repeated-Measures ANOVA for Response Time.

Study		*SS*	*df*	*F*	*P*
1	Alg.	419	3	21.92	4.45e-11
2	Alg.	207	3	10.61	3.24e-06

Table 10: Repeated-Measures ANOVA for Perceived Performance.

Study		*SS*	*df*	*F*	*P*
1	Alg.	184	3	16.37	8.70e-09
2	Alg.	72.1	3	9.64	9.89e-06

Conversely, compared with the fully observable domain, the Time Delay interaction method was the most improved method. Whereas it consistently rated as the worst interaction method in a fully observable domain, once the domain became more complex and only partially observable, small time delays were considered more acceptable, and from the intelligence ratings, perhaps even expected of a more intelligent agent. In the partially observable domains, the Time Delay interaction rates very similarly to the 1-Step and 5-Step interaction methods.

Frustration levels with the agents were lower for the partially observable domains, but much higher overall once penalties are introduced. We see that the Time Delay method actually caused the least frustration in a domain with penalties. In terms of transparency, the 5-Step method consistently scored the highest in every study, indicating that the participant could understand the agent's actions. All methods of interaction, except for Probabilistic, resulted in equal perceived performances. However, once penalties were introduced into the partially observable en-

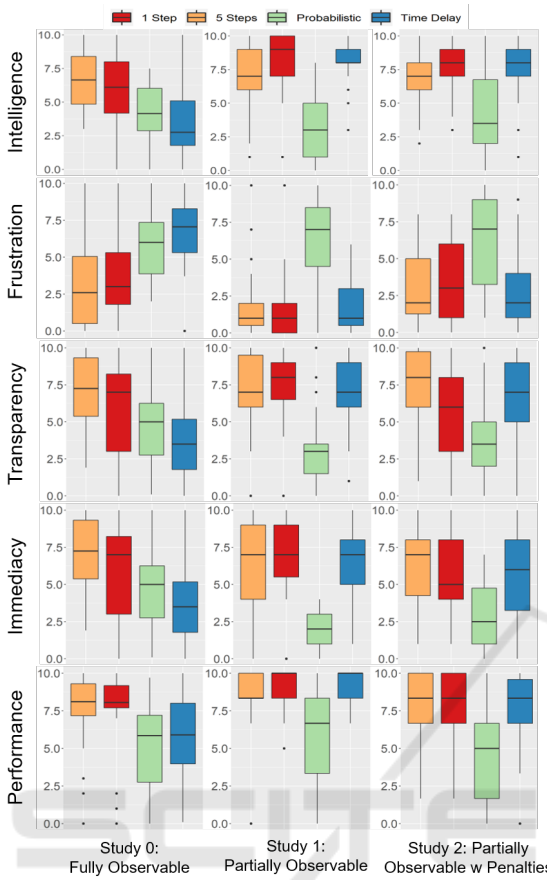

Figure 7: Box plots showing human experience measure for interaction methods.

vironment, the variance of each agent's perceived performance increased.

In terms of the perceived intelligence of the agents, the 5-Step method is considered the most intelligent in fully observable domains, while the 1-Step method is considered the most intelligent in partially observable domains. However, once penalties are introduced into the partially observable environment, the Time Delay method is considered equally intelligent to the 1-Step method, probably because it performs the same way as the 1-Step method, just slower. The perceived intelligence of the agent is analyzed more closely in Figure 8, which depicts the participants' rankings of the agents from most to least intelligent at the end of the experiment. The 5-Step method is considered one of, if not the most, intelligent agent in the domains with penalties. The 1-Step interaction method was considered the most intelligent in the domain without penalties. This is interesting because in the individual ratings of the agents' intelligence shown in Figure 7, the 1-Step agent is considered one of, if not the most, intelligent agent in both of the partially observable domains. This indicates that the agent that employed the 5-Step interaction method left an overall more positive impression on the participant after seeing all 4 interaction methods, even though individually, the 1-Step agent was considered the most intelligent. It can also be seen that the Time Delay method steadily gained popularity from study to study as the domain increased in complexity.

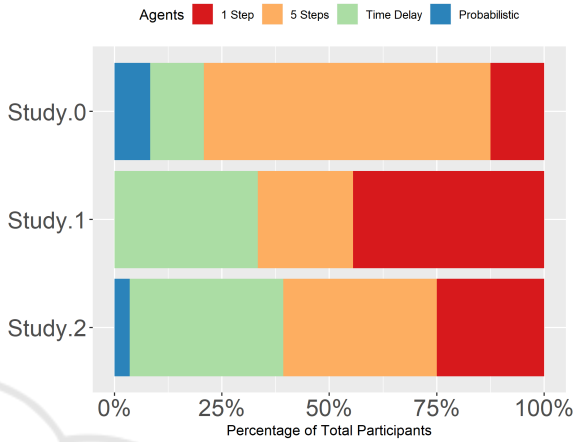

Figure 8: Rankings of user's perceived intelligence of the agent based on interaction method.

In the final questionnaire, participants were also asked in a free-response question why they ranked the agents the way they did. According to the responses, several aspects of the agent's interaction impacted the participants' perceived intelligence of the agent and their own level of frustration in teaching it. There were many commonalities across the participants' responses; the percentage of participants who mentioned certain features in their responses are shown in Tables 11 and 12, corresponding with Study 1 and Study 2, respectively.

Table 11: Percentage of participants who mentioned in their long responses certain features that contributed to the user experience of Frustration and Perceived Intelligence in Study 1: Partially Observable Domain.

Feature	Frustration	Intelligence
Compliance	67	67
Effort	24	14
Frustration	10	10
Immediacy	33	38
Improvement	0	0
Less Instruction	29	24
Memory	10	24
Randomness	33	33
Repeating Myself	0	0
Transparency	33	43

Table 12: Percentage of participants who mentioned in their long responses certain features that contributed to the user experience of Frustration and Perceived Intelligence in Study 2: Partially Observable Domain with Penalties.

Feature	Frustration	Intelligence
Compliance	50	47
Effort	7	0
Frustration	13	4
Immediacy	47	30
Improvement	7	17
Less Instruction	13	27
Memory	3	17
Randomness	27	40
Repeating Myself	7	4
Transparency	23	7

In the fully observable domain, the top aspects cited by participants as impacting their overall experience teaching the agent were: 1) compliance with advice (whether the agent followed the person's advice), 2) response time (how quickly the agent followed advice), 3) quantity of instruction required, and 4) randomness (whether the agent was perceived to act in an unreliable manner) (Krening and Feigh, 2019). Based on the responses in Tables 11 and 12, the top aspects contributing to the user's experience are the same.

4 CONCLUSIONS

As artificial intelligence becomes more prevalent in everyday life, the experience of teaching ML agents must become accessible and easy. A key aspect of enabling ML to be accessible and easy to train will be to understand what interaction mechanisms are most appropriate and experientially pleasurable. The work here begins to provide an understanding of the impact of different interaction methods on human experiences in domains of varying complexity. In all domains, it is recommended to maximize an agent's adherence to advice and reliability in behavior. In partially observable domains, it appears to be both allowable and even recommended to incorporate a short time delay to give the teacher some time to make a decision regarding the agent's next action. The decision of generalizing movement through time is left to the algorithm designer since the user perception and objective performance of these interaction methods vary in partially observable domains.

ACKNOWLEDGMENTS

We thank Dr. Sam Krening for sharing data from Study 0 and for her support. This work was funded under ONR grant number N000141410003. All opinions and conclusions expressed are solely those of the authors.

REFERENCES

Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. (2014). Power to the people: The role of humans in interactive machine learning. volume 35, pages 105–120.

Frazier, S. and Riedl, M. (2019). Improving deep reinforcement learning in minecraft with action advice. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pages 146–152.

Klein, N. and O'Brien, E. (2018). People use less information than they think to make up their minds. volume 115, pages 13222–13227. National Academy of Sciences.

Krening, S. (2018). Newtonian action advice: Integrating human verbal instruction with reinforcement learning. In *arXiv, arXiv:1804.05821*.

Krening, S. and Feigh, K. M. (2018). Interaction algorithm effect on human experience with reinforcement learning. volume 7, New York, NY, USA. Association for Computing Machinery.

Krening, S. and Feigh, K. M. (2019). Effect of interaction design on the human experience with interactive reinforcement learning. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, page 1089–1100. Association for Computing Machinery.

Lin, Z., Harrison, B., Keech, A., and Riedl, M. O. (2017). Explore, exploit or listen: Combining human feedback and policy model to speed up deep reinforcement learning in 3d worlds. *ArXiv*, abs/1709.03969.

Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287.

Thomaz, A. L. and Breazeal, C. (2006). Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, page 1000–1005. AAAI Press.

Thomaz, A. L., Hoffman, G., and Breazeal, C. (2005). Real-time interactive reinforcement learning for robots. In *In: Proc. of AAAI Workshop on Human Comprehensible Machine Learning*.