Using Informative Behavior to Increase Engagement in the TAMER Framework

Guangliang Li
University of Amsterdam
Amsterdam, Netherlands
g.li@uva.nl

Hayley Hung
University of Amsterdam
Amsterdam, Netherlands
h.hung@uva.nl

Shimon Whiteson
University of Amsterdam
Amsterdam, Netherlands
s.a.whiteson@uva.nl

W. Bradley Knox
MIT Media Lab
Massachusetts, USA
bradknox@mit.edu

ABSTRACT

In this paper, we address a relatively unexplored aspect of designing agents that learn from human training by investigating how the agent's non-task behavior can elicit human feedback of higher quality and quantity. We use the TAMER framework, which facilitates the training of agents by human-generated reward signals, i.e., judgements of the quality of the agent's actions, as the foundation for our investigation. Then, we propose two new training interfaces to increase active involvement in the training process and thereby improve the agent's task performance. One provides information on the agent's uncertainty, the other on its performance. Our results from a 51-subject user study show that these interfaces can induce the trainers to train longer and give more feedback. The agent's performance, however, increases only in response to the addition of performance-oriented information, not by sharing uncertainty levels. Subsequent analysis of our results suggests that the organizational maxim about human behavior, "you get what you measure"—i.e., sharing metrics with people causes them to focus on maximizing or minimizing those metrics while de-emphasizing other objectives— also applies to the training of agents, providing a powerful guiding principle for human-agent interface design in general.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Performance, Human Factors, Experimentation

Keywords

reinforcement learning; human-agent interaction

1. INTRODUCTION

As autonomous agents become more sophisticated, they are likely to be an increasingly integral part of our daily lives. How well they are accepted by human users will depend heavily on whether they can interact effectively with them, particularly those without expertise in autonomous agents or even technology in general. Therefore, there is a great need to improve our understanding of how to develop more sophisticated interfaces that facilitate this interaction.

One critical form of human-agent interaction occurs when agents learn with human assistance. Many approaches (e.g., learning from demonstration [2, 3, 26], giving advice to reinforcement learners [12, 21], and learning from human feedback [22, 23, 27, 28]) have been developed to enable such interaction. One such approach, the TAMER framework [13, 15], enables humans to train agents to perform tasks by giving scalar feedback signals in response to the agent's behavior. A TAMER agent creates a predictive model of human feedback and myopically chooses the action at each time step that it predicts will receive the highest feedback value. Similar to any algorithm that enables humans to teach agents, a TAMER agent's learned performance depends critically on the quality and quantity of feedback that the human provides. Here, we use TAMER as a platform to investigate whether changes to the interaction interface can increase the trainer's involvement by measuring the quality and quantity of feedback and the duration of training. Such changes can also apply to other learning methods e.g., learning from demonstration.

Our approach is motivated by the notion that, similar to a human student and teacher, the interactions between a TAMER agent and its human trainer should ideally be bidirectional. For example, in addition to the human giving feedback to the agent, the agent should also give feedback to the human to inform them about its progress and indicate the kind of human feedback that would be most useful. Thus, not only should the human teach the agent how to complete the task, the agent should also influence the human to teach it as effectively as possible.

In this paper, we investigate this approach within TAMER, focusing on the impact of the interaction interface's design. Specifically, we study the use of two new interfaces for the TAMER framework. In the first, the *uncertainty-informative* interface, the agent informs the human of its uncertainty about the actions it selects, in the hope that this motivates the human to reduce that uncertainty by focusing feedback on the most needed areas. In the second, the *performance-informative* interface, the agent informs the human about its current performance in the task relative to its earlier performance, which we expect will motivate the human to give the

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013),* Ito, Jonker, Gini, and Shehory (eds.), May 6–10, 2013, Saint Paul, Minnesota, USA.

feedback needed to further improve this performance. We hypothesize that these informative behaviors will cause the human to (i) train longer, (ii) give more feedback, and (iii) that the agent's performance will improve as a result.

To test these hypotheses, we compared the agents with informative behaviors to the original TAMER agent in a human-user study with 51 subjects. Our results indicate that both informative interfaces increase the duration of training and the amount of feedback provided, with the uncertainty-informative interface generating the most feedback. The results also show that whereas the performance-informative interface improves the performance of the TAMER agent, the uncertainty-informative interface unexpectedly reduces its performance.

For each interface, we also used principal component analysis to visualize the distribution of states that the agent visited and in which feedback was received. The analysis highlights very different feedback behavior where, in the performance-informative condition, feedback tends to be given only in concentrated and similar parts of the state space, while, in the other conditions, feedback is given much less selectively.

Altogether, these results not only offer new insights into TAMER, they also highlight the importance of interface design—a previously under-emphasized aspect of agent training using humans —by providing evidence of its influence on human training behavior. Furthermore, the results fit a pattern observed in organizational behavior research that follows from the adage "you get what you measure" [7]. That is, sharing behavior-related metrics will tend to make a human attempt to improve their score with respect to that metric. To our knowledge, this is the first time this phenomenon has been observed in the behavior of trainers for agent learning.

The rest of this paper begins with a review of related work in Section 2 and provides background on TAMER in Section 3. Section 4 presents the proposed interfaces, Section 5 describes the experimental setup, and Section 6 reports and discusses the results. Finally, Section 7 discusses future work and concludes.

2. RELATED WORK

Here, we discusses the research most related to our approach, namely work on learning from demonstration and learning interactively from human feedback as well as a discussion of past approaches in which the agent acts to explicitly affect the behavior of its human teacher.

2.1 Learning from Demonstration

In *learning from demonstration*, a human assists the agent's learning by demonstrating the task via teleoperation or shadowing (the learner records the task execution using its own sensors and attempts to match or mimic the teacher's motion as the teacher executes the task). The agent learns a behavior policy from these demonstrations that reproduces and generalizes the demonstrated behavior [3]. For example, *apprenticeship learning* [1], is a form of learning from demonstration in which an expert's demonstrations are used to estimate a hidden reward function.

In most learning by demonstration systems, the agent is a passive recipient of the demonstrations and cannot actively gather data to influence the learning process. As a result, it can only imitate the teacher's behavior and its performance is thus limited by the teacher's own skill level. However, methods have also been developed to further improve the agent's ability to learn from such a teacher. For example, Argall et al. [2] propose an approach wherein learning from demonstration is coupled with critiques by the human of the agent's performance. Chernova and Veloso [9] propose to make the agent's learning active by using a framework in which the agent, on the basis of its confidence in what it has learned, can request a specific demonstration from the human and the human can correct the agent's mistakes.

These approaches are similar in motivation to ours, in that they seek human-agent interfaces that aid agent learning. In addition, the approach of Chernova and Veloso is related to the uncertainty-informative interface we propose in Section 4.1, in that the agent's uncertainty is used to guide this interaction. However, the focus of these methods is different but complementary to ours. These two studies are concerned with the impact of enabling the agent to query the trainer and of enabling the trainer to give corrective feedback after observing behavior learned from demonstration; we investigate how trainer behavior—and resultant agent performance—is influenced by the specific stream of information that the agent shares with the trainer.

2.2 Learning from Human Feedback

In this learning category, a human trains the agent by providing feedback signals that evaluate the quality of the agent's actions and state transitions [15, 22, 23, 27]. As Knox and Stone write, "in contrast to the complementary approach of learning from demonstration, learning from human [feedback] employs a simple task-independent interface, exhibits learned behavior during teaching, and, we speculate, requires less task expertise and places less cognitive load on the trainer" [17].

One of the earliest attempts to train artificial agents in this way is based on *clicker training* [5], a form of animal training in which the sound of an audible device such as a clicker or whistle is associated with a primary reinforcer such as food and then used as a reward signal to guide the agent towards desired behavior. In addition, Thomaz and Breazeal [28] propose a reinforcement-learning agent that combines the standard Q-learning algorithm [31] with a separate interaction channel by which the human can give the agent feedback. The agent aims to maximize its total discounted reward, where the human's feedback is treated as additional reward that supplements the environmental reward.

The TAMER framework [15] allows an agent to learn from human reward signals instead of environmental reward. These reward signals are provided by a human trainer who observes and evaluates the agent's behavior while the agent is trying to perform the task. The primary differences between the TAMER framework and other algorithms for learning from human feedback are that TAMER creates a predictive model of human reward, explicitly addresses delay in the delivery of human reward signals, and chooses actions that its human model predicts will elicit maximal reward through fully myopic valuation, considering only reward caused by its immediate action. General myopia is a feature of all past algorithms for learning from human feedback and received empirical support in recent work [17], but TAMER is unique in that it is fully myopic (in reinforcement learning terminology, it values future reward with $\gamma = 0$).

In the TAMER+RL framework [16, 18], the agent learns from both human and environmental feedback, which can

lead to better performance than learning from either alone. This can be done sequentially (i.e., the agent first learns from human feedback and then environmental feedback) [16] or simultaneously (i.e., the agent learns from both at the same time), allowing the human trainer to provide feedback at any time during the learning process [18].

2.3 Agent Behavior that Influences the Trainer

This paper builds on the TAMER framework but focuses on how to improve training through the interface design. In particular, past work on TAMER and TAMER+RL only communicates the agent's action and environmental state to the trainer and does not empirically analyze what agent information *should* be communicated to elicit training of higher quality or longer duration. In this paper, we perform such a comparative analysis.

Our approach is thus related to work in human-robot interaction on *transparent* learning mechanisms, where facial expressions and body language are used to express the robot's learning state and solicit feedback from someone [8, 20, 29, 30]. Similarly to our approach, the agent provides the human with information about its learning process. However, our work is the first to consider this within the TAMER framework and provides the first analysis of how manipulating the information the agent provides can affect the trainer's behavior. Furthermore, our empirical user study provides evidence that such informative behavior increases the trainer's feedback quantity and quality.

In this paper, we frame the information sharing of an agent's interactive interface as a form of communicative behavior. In related work, this information sharing was achieved directly through the agent's task-focused behavior in an experiment in which the agent deviates from its greedy behavior—intentionally choosing actions it believes to be sub-optimal—whenever the trainer's recent feedback is sparse, in effect punishing the trainer for inattentiveness [14]. The results showed that, in comparison to TAMER agents that simply act greedily, these manipulative agents elicited a higher overall frequency of feedback and yet performed worse.

3. BACKGROUND

This section briefly introduces the TAMER framework and the Tetris platform used in our experiment.

3.1 TAMER Framework

In the TAMER framework, the agent strives to maximize the reward caused by its immediate action, not a discounted sum of future rewards. The intuition for why an agent *can* learn to perform tasks by such myopic valuation of reward is that human feedback can generally be delivered with small delay—the time it takes for the trainer to assess the agent's behavior and deliver feedback—and the evaluation that creates a trainer's reward signal carries an assessment of the behavior itself, with a model of its long-term consequences in mind. Recent analysis indicates that agents *should* act myopically in episodic tasks [17].

The TAMER framework is built to solve a variant of Markov decision processes, (i.e., model of a sequential decision-making problem commonly used in reinforcement learning [24]) in which there is no reward function encoded before learning. Instead, the agent learns a function $\hat{H}(s,a)$ that approximates the expectation of experienced human reward, $H : S \times A \rightarrow \Re$. Given a state s, the agent myopically chooses the action with the largest estimated expected reward, $\arg\max_a \hat{H}(s,a)$. The trainer observes the agent's behavior and gives reward corresponding to its quality.

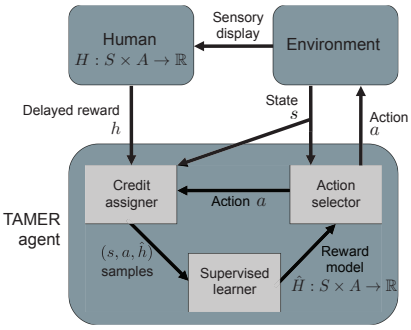


Figure 1: Interaction in the TAMER framework (reproduced from [13]).

The TAMER agent treats each observed reward signal as a label for the previous (s,a), which is then used as a supervised learning sample to update the estimate of $\hat{H}(s,a)$. In this paper, the update is performed by incremental gradient descent; i.e., the weights of the function approximator specifying $\hat{H}(s,a)$ are updated to reduce the error $|r - \hat{H}(s,a)|$, where r is the reward observed after taking action a in state s. Figure 1 illustrates interaction in the TAMER framework.

In our experiments, we used the original baseline TAMER interface presented in [15] as a control condition.[1] In this interface, feedback is given via keyboard input and attributed to the agent's most recent action. Each press of one of the feedback buttons registers as a scalar reward signal (either -1 or +1). This signal can also be strengthened by pressing the button multiple times. The TAMER learning algorithm loops by taking an action, sensing reward, and updating $\hat{H}$.

3.2 Tetris Platform

Tetris is one of the most popular computer games in the world. Although it has simple rules, it is a challenging problem for agent learning because the number of states required to represent all possible configurations of the Tetris board is extremely large [10]. In the TAMER framework, the agent uses 46 state features—including the 10 column heights, 9 differences in consecutive column heights, the maximum column height, the number of holes, the sum of well depths, the maximum well depth, and the 23 squares of the previous 23 features [13]—to represent the state observation. The input to $\hat{H}$ is 46 corresponding state-action features, the difference between state features before and after a placement. Tetris is an excellent platform for investigating how humans and agents interact during agent learning because it is a fun game that is familiar to most human trainers. We use an adaptation of the RL-Library implementation of Tetris.[2]

Like other implementations of Tetris learning agents (e.g.,[4, 6, 25]), the TAMER agent chooses from possible final placements of pieces upon the stack of previously placed pieces, instead of controlling atomic rotations and left/right movements. Even with this simplification, playing Tetris remains a complex and highly stochastic task. For the standard

[1]Beyond the interface, the agents are all identical to those in the control condition of [14], which use more state features than in [15].

[2]`library.rl-community.org/wiki/Tetris_(Java)`

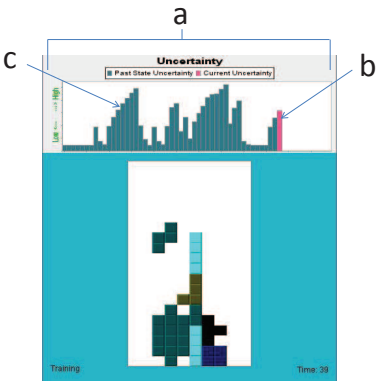


Figure 2: The uncertainty-informative interface: (a) the uncertainty graph window, (b) the current uncertainty (pink bar), and (c) the uncertainty of past actions (dark blue bar).

20×10 board size that we use, the state space is greater than 2^{200}.

To eliminate training effects that can be caused by perception speed, this interface allows the trainer to adjust the falling speed of each piece by pressing the '+' and '–' buttons respectively. Button 'z' is used for negative reward, and button '/' is used for positive reward. To inform the trainer that he/she is giving feedback, the game screen flashes blue and red for positive and negative reward, respectively. Moreover, the trainer can also pause the game with the space bar, and continue training with button '2'. The interface is a Java applet that runs in the trainer's browser.

4. INFORMATIVE INTERFACES

In this section, we propose two variations on the baseline TAMER interface described above that each display additional information about the agent's performance history or internal processes. These variations are motivated by the notion that the interaction between the trainer and the agent should be consciously designed to be bidirectional, where the agent gives the trainer informative and/or motivating feedback about its learning process. Our intuition is that such feedback will help keep the trainer's involvement in the training process and empowers them to offer more useful feedback. More objectively, we hypothesize that doing so will increase the quantity of the trainer's feedback and improve the agent's task performance.

4.1 Uncertainty-Informative Interface

The first variation is the *uncertainty-informative* interface, in which the agent indicates to the trainer its uncertainty about the action it selects. We hypothesize that doing so will motivate the trainer to reduce uncertainty by giving more feedback and enable them to focus that feedback on the states where it is most needed. To implement this interface, we added a dynamic bar graph above the Tetris board that shows the agent's uncertainty, as shown in Figure 2.

Many methods are possible for measuring the agent's confidence of action selection. For example, the confidence execution algorithm in [9] uses nearest neighbor distance from demonstrated states to classify unfamiliar, ambiguous states. Since we are primarily interested in how a trainer's perception of the agent's uncertainty of an action affects training behavior, we applied a simple uncertainty metric that we expected to maximize the amount of feedback given. While more sophisticated uncertainty metrics could also be used, optimizing this metric is not the focus of our study.

Our approach considers an agent to be more certain about its action in a state if it has received feedback for a similar state. We calculate the weighted sum of the distance in feature space from the current state to the k nearest states that previously received feedback, yielding a coarse measure of the agent's uncertainty about the current action. Thus, we define the uncertainty U of the current state s_c as

$$U(s_c) = \Sigma_{i=1}^{k} w_i d_i, \quad (1)$$

where d_i is the Euclidean distance between the current state s_c and the i-th closest state s_i in the set of all states wherein the agent received feedback:

$$d_i = \sqrt{\Sigma_{j=1}^{n}(s_{ij} - s_{cj})^2}, \quad (2)$$

where s_{ij} is the value of the j-th state feature of state s_i. We chose the weights w_i (where $\Sigma_{i=1}^{k} w_i = 1$) to approximate an exponential decay in the ranking of farther neighbors. In our experiments, $k = 3$ and $(w_1, w_2, w_3) = (0.55, 0.3, 0.15)$.

While the human is training the agent, the graph shows both the uncertainty of the current state (pink rightmost bar) and past states (dark blue bars), as illustrated in figure 2. In the graph window, the uncertainty at each time step (each time a piece is placed) is shown from left to right chronologically. For the first game, no bar is shown until the agent has received feedback three times. Then, before each piece is placed, the agent shows its current uncertainty for the action it is about to make. If the trainer gives feedback, the value of the current uncertainty is modified according to Equation 1 and the new uncertainty is visualized as a dark blue bar after the piece is placed. Meanwhile, a new pink bar appears to the right of it, showing the uncertainty of the new placement. Since the graph window can only show up to 60 time steps, it is cleared when the current time step is a multiple of 60, and the bar showing the new uncertainty is shown from the left side again. The vertical axis is labeled only with "high" and "low" so that trainers focus on relative differences in uncertainty, not absolute values. To keep the changes in uncertainty visible, the interface starts with a fixed maximum uncertainty value; if the height of the bar exceeds this maximum, the ceiling value of the vertical axis is correspondingly adjusted. When the height of the bar exceeds the ceiling value of the vertical axis, the ceiling value is automatically adjusted to fully show the highest bar.

4.2 Performance-Informative Interface

The second variation is the *performance-informative* interface, in which the agent indicates to the trainer its performance over past and current games. We hypothesize that explicitly displaying performance history will increase the trainer's motivation to improve the agent's performance, thus leading to more and higher quality feedback. To implement this interface, we again added a dynamic bar graph above the Tetris board, as shown in Figure 3. In this case, however, each bar indicates the agent's performance in a whole Tetris game. Since clearing a line reduces the stack height and in turn gives the agent the opportunity to clear more lines, we quantified the performance of the agent by the number of lines cleared. This metric is both intuitive for the trainer to understand and fits with past work on agents that learn to play Tetris [6, 25]. The interface was designed to look very similar to the uncertain-informative interface to avoid confounding factors. During training, the graph shows

the agent's performance (i.e., lines cleared per game) during past and current games, ordered chronologically from left to right, so the trainer can keep track of the agent's progress.

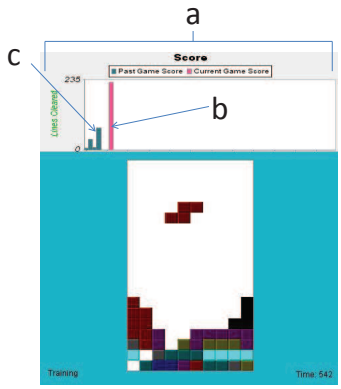


Figure 3: The performance-informative interface: (a) performance graph window, (b) current game performance, and (c) performance of past games.

During the first game, after the first line is cleared, the graph shows a pink bar at the left side of the graph window, representing the number of lines cleared so far. When a game ends, its corresponding bar becomes dark blue and any new lines cleared in the new game are visualized by a pink bar to its right. As in the uncertainty-informative interface, the window is cleared after it is filled with 60 bars—games in this case—and new bars appear from the left. The vertical axis is labeled 'Lines Cleared' and is initially bounded between 0 and 10. When the number of lines cleared exceeds the axis' upper bound, the limit is increased by 25 while all prior performance is in the range $[0, 100]$ and the height of all bars are adjusted accordingly. Subsequently, the upper bound is increased by 50 for the range $(100, 1000]$, 100 for the range $(1000, 10000]$, and 1000 for greater than 10000.

5. EXPERIMENTAL SETUP

To maximize the diversity of subject recruitment, we deployed the Tetris game on the internet. 70 participants from more than 10 countries were recruited by email, a Facebook page and flyer and poster advertisements. Their ages ranged from 19 to 63 and included both males and females. Some had backgrounds in AI or related fields while others had little knowledge of computer science; at least 8 had no programming skills. Of the 38 subjects who filled in the post-experiment questionnaire, 9 were from the Netherlands, 8 from China, 7 from Austria, 3 from Germany, 2 each from the USA, Italy, and Greece, and one each from the UK, Belgium, Japan, Canada, and Turkey.

In the experiment, all the participants were told "In this experiment you will be asked to train an agent to play Tetris by giving positive and negative feedback" in the instructions. The instructions also described how to give feedback and, for the appropriate conditions, explained the agent's informative behavior. The subjects were divided evenly and randomly into the three conditions. However, data from 19 of the recruited subjects was removed because the subjects registered but never started training or used the wrong ID when returning to train their agent (so they trained multiple agents). Thus, the rest of this paper analyzes the results from the remaining 51 participants. The experimental details of each condition are described below:

- **Control Condition**: 16 participants trained the agent without seeing any informative behaviors using the baseline TAMER interface described in Section 3.1.
- **Uncertainty-Informative Condition**: 19 participants trained the agent using the interface described in Section 4.1.
- **Performance-Informative Condition**: 16 participants trained the agent using the interface described in Section 4.2.

The participants were encouraged to train the agent as many times as they liked during 7 days. We recorded the state observation, actions, human rewards, lines cleared, the absolute start time, speed of each time step, lines cleared per game, and number of training sessions.

We also investigated the correlation between the trainer's training behavior and certain characteristics of the trainer's personality. We hypothesized that for the uncertainty and performance-informative conditions, a more empathetic and competitive trainer respectively for each condition will spend more time on training, leading to higher performing agents.

To validate these hypotheses, we designed a questionnaire to measure the trainer's feelings about the agent, the training process, and the personalities of the trainer. We used a 5-point scale composed of bipolar adjective pairs: 7 to test the trainer's feelings about the agent, and 10 for the training process. Participants were also given a self-report scale including 13 items from the Empathy Quotient (EQ) [19] to measure the trainers' willingness to interpret the informative behavior and 14 items as a measure of competitiveness, which were adapted from the Sport Orientation Questionnaire [11]. 8 filler items were also included to minimise potential bias that can be caused by the trainers second-guessing what the questionnaire was about (four of them were from the Introversion-Extroversion Scale and another four from EQ [19]).

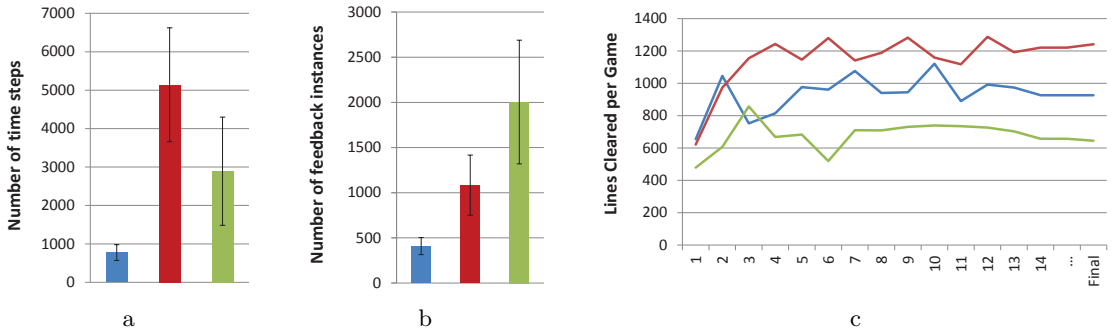
6. RESULTS AND DISCUSSION

This section presents and analyzes the results of our human-user study. All reported p values were computed via a two-sample t-test. Since each hypothesis specifies a one-directional prediction, a one-tailed test was used. Additionally, an F-test was used to assesses whether the dependent variables for each condition have equal variances, since the two-sample t-test is calculated differently if the difference in variance for the two samples is significant (i.e., $p < 0.05$).

6.1 Training Time

We hypothesized that both conditions would increase the time spent on training compared to the control condition. We found that both informative behaviors did engage the trainers for longer, in terms of both absolute time and number of time steps. In mean absolute time, trainers in the performance-informative condition spent 220% more time on training ($t(17) = 1.74, p < 0.02$), as did trainers in the uncertainty-informative condition, who spent 128% longer ($t(21) = 1.72, p = 0.05$). In mean time steps—a metric unaffected by the player's chosen falling speed—for the performance-informative and uncertainty-informative conditions, the number of time steps spent training the agent were 565% ($t(16) = 1.75, p < 0.01$) and 274% ($t(19) = 1.73, p = 0.076$) more than for the control condition, as shown in Figure 4a.

To measure the amount of feedback given, we counted the number of times a feedback button was pressed, comparing each experimental condition to the control. As shown in Figure 4b, in the performance-informative condition, 165%



a b c

Figure 4: Number of time steps trained (a), number of feedback instances given during training (b) and offline performance per cumulative interval for the three conditions (c). Control condition: blue, performance-informative condition: red, uncertainty-informative condition: green.

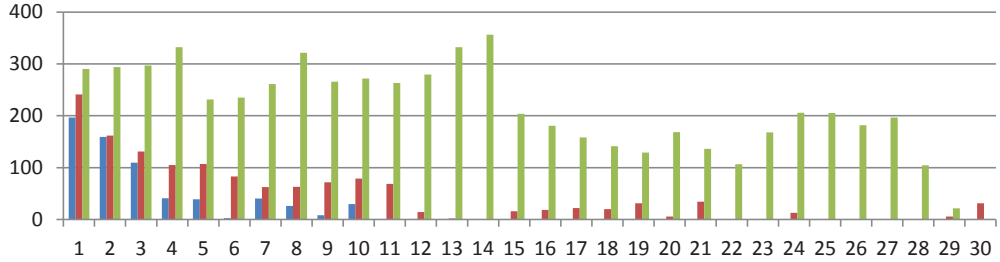


Figure 5: Number of feedback instances over the first 30 intervals. Each interval is 200 time steps. Control condition: blue, performance-informative condition: red, uncertainty-informative condition: green.

more feedback is given ($t(17) = 1.74, p = 0.034$), whereas in the uncertainty-informative condition, 390% more feedback than in the control condition ($t(19) = 1.73, p = 0.016$). In addition, the number of time steps with feedback (irrespective of the number of feedback instances at each time step) for both informative conditions was significantly more than the control condition (performance-informative: $t(19) = 1.73$, $p < 0.02$; uncertainty-informative: $t(19) = 1.73, p < 0.025$).

Figure 5 shows how feedback was distributed over the first 30 intervals, which each interval contains 200 time steps. The longest training time is about 100 intervals; because of limited space, we show only the first 30 intervals. After 30 intervals, the subjects in the uncertainty-informative condition still give more feedback than the other conditions. Trainers in both informative conditions gave feedback for much longer than those in the control condition. Most notably, trainers in the uncertainty-informative condition gave a strikingly large amount of feedback, even during later intervals, with a much slower fall-off than the other conditions.

Thus, our results clearly suggest that informative behavior can significantly increase the amount of feedback given and time spent training, suggesting better involvement.

6.2 Performance

We also hypothesized that the trainers' increased involvement would lead to improved performance by the agents. To test this, we first examined how the agents' performances varied over time. Because the duration of a game varies significantly depending on the quality of the trained policy, we saved each agent's policy at regular time intervals. We used intervals of 200 time steps to get a picture of reasonable resolution of the agents' progress. Then, we tested the saved policy of each agent off-line for 20 games. Figure 4c shows the resulting mean performance averaged across games, and then agents, for the first 14 intervals, as well as the final off-line performance, after all training was complete. If an agent's training stopped before others in the same condition, then in later intervals its final average off-line performance was used to compute the average for that condition.

Compared with the control condition (average final off-line performance is 926.5 lines), agents in the performance-informative condition learned best (1241.8 lines, $t(30) = 1.70, p = 0.267$), while those in the uncertainty-informative condition learned worst (645.1 lines, $t(33) = 1.69, p = 0.24$). While the differences are not significant (perhaps due to the small sample size), Figure 4c shows that the uncertainty-informative condition performs consistently worse than the other conditions while the performance-informative condition performs consistently better.

These differences in performance did not always match our hypotheses. As expected, agents in the performance-informative condition had better performance than the control condition. However, whereas the uncertainty-informative condition generated significantly more feedback than any other condition, the resulting agents had the worst performance of all three conditions.

6.3 State Feedback Behavior

The main surprise in the results presented in Sections 6.1 and 6.2 is that, while the uncertainty-informative condition elicited the most feedback, the resulting agents performed substantially worse. This result is especially puzzling given that the performance-informative condition also elicited more feedback than the control condition but generated agents with better performance. We note that the learning algorithm had no access to uncertainty or performance metrics, and the interfaces differ only in their feedback to the trainer. Therefore, differences in agent performance result only from the feedback given by the agent.

We hypothesized that this discrepancy could be because the trainers in the performance-uncertainty condition were influenced by feedback that was better aligned with the goal of the game. In other words, because they were shown the agent's performance, they were more motivated to train the

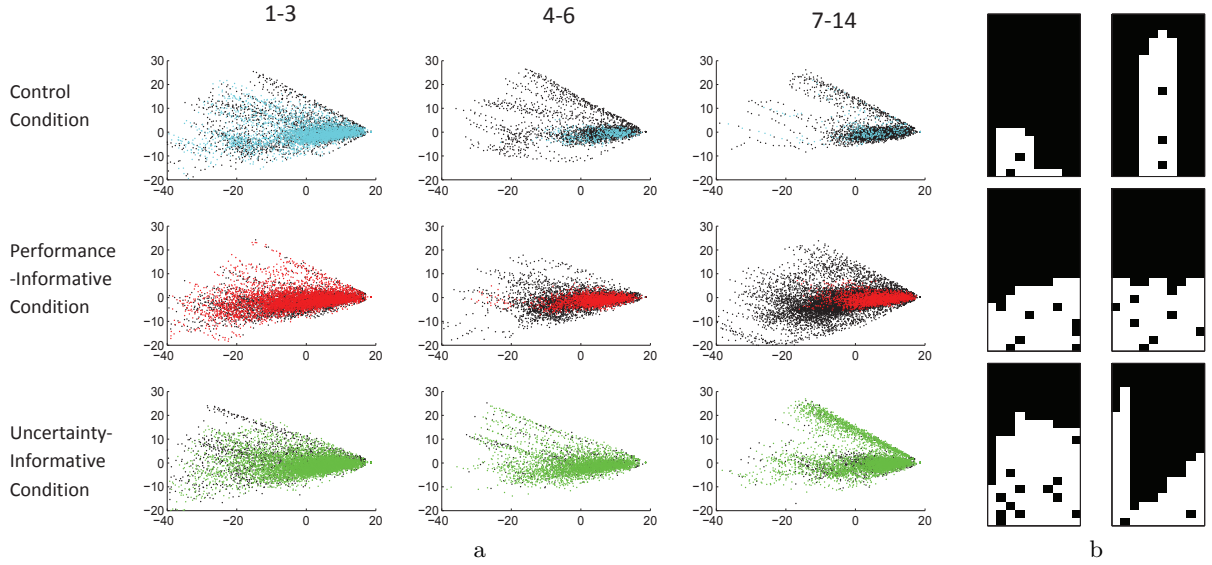


Figure 6: a): Distribution of states along the first(horizontal axis) and second (vertical axis) principal components, with and without feedback. The columns show intervals 1-3, 4-6, and 7-14, from left to right, respectively. b): the left and right columns shows example stacks along the first and second principal components respectively, ordered from corresponding positive (top) to negative (bottom) component values.

agent to maximize performance. In contrast, we suspect that trainers in the uncertainty condition were distracted from this goal by the agent's informative behavior. That is, while they trained for longer and gave more feedback than trainers in the control condition, they were more focused on giving feedback that would reduce the uncertainty bar, rather than giving feedback that would maximize performance. In an effort to test this hypothesis, we analyzed how the state of the game itself might have influenced the feedback given. To this end, we analyzed all the states visited by all of the agents in every condition using *principal component analysis* (PCA). Then, within the feature space created by the first two principal components, we examined how the distribution over states in which feedback was given differed across time and across the three conditions.

As shown in Figure 6a, we divided the training process in each condition into 4 sections chronologically: intervals 1-3, 4-6, 7-14, and 15 and higher. For brevity, we do not show plots for intervals 15 and greater; at this point, all trainers in the control conditions had stopped; plots for the other two conditions are qualitatively similar to those for intervals 7-14. Again, each interval is composed of 200 time steps. From top to bottom, the three rows show the control, performance and uncertainty-informative conditions. We plotted the projection of the visited states onto the first two principal components of the data. Since there were many more states without feedback (shown in black), for visualization, they were overlaid with those that received feedback, which were colored according to their corresponding condition: control in blue, performance-informative in red and uncertainty-informative in green. The proportion of variance explained by the first and second components are 45.01% and 9.49%, respectively.

Figure 6a shows that, in the initial stage (intervals 1-3), informative behaviors seem to have little influence on the distribution of states with feedback. However, in all intervals thereafter, the performance-informative behavior appears to keep the trainer focused on giving feedback in states in the center of the second principal component (around 0), while the uncertainty-informative behavior receives feedback in a much wider range of states along the second principal component. Note that in the uncertainty-informative condition, not all trainers exhibited the broader feedback behavior; trainers that gave more focused feedback (like the performance-informative users) had the better performing agents. If feedback is given to states which do not benefit or even harm learning, this discrepancy may explain the difference in performance between performance and uncertainty-informative conditions. On closer inspection of the distribution of states with and without feedback, we observed that, in all cases, the distributions were unimodal.

Inspecting the coefficients of the principal eigenvectors, we observed that the state features corresponding to the height of the stack and the number of holes contributed most to the first principal component, and so the narrow point at the far right of the space is representative of the start of each game when the stack height is 0. For the second principal component, positive weights were found for columns 1, 2, 9, and 10, of the tetris board, while the features representing the column heights 4, 5, 6, and 7 were negatively correlated. This is seen more clearly in Figure 6b, where along the first principal component, the overall height of the Tetris board is gradually increasing while keeping roughly flat. For the second principal component, the contour of the Tetris board is changing from n-shaped to u-shaped from the top to the bottom. Combining with Figure 6a, we observe that, in the uncertainty-informative condition, a lot of feedback was given to n-shaped states, which intuitively would not benefit learning, especially in intervals 7-14. This may explain the poor performance in the uncertainty-informative condition, which indirectly supports our second hypothesis.

As well as providing an insight for TAMER, these results also align with what could be expected from the maxim, "you get what you measure"; i.e., people often try to optimize the metrics you show them while deemphasizing others. In our case, measuring, or informing users about performance increased performance, and measuring uncertainty reduced uncertainty, through increased feedback, but reduced per-

formance. The notion that "you get what you measure" has been discussed extensively in organizational literature (e.g., metrics for software development teams [7]), but we believe this paper is the first to find evidence that suggests the concept applies to the design of interactive interfaces for training agents. Consequently, understanding the influence of metric-sharing on human behavior could be a powerful guiding principle in the design of interactive interfaces for training agents, though more investigation is needed to judge its general applicability.

7. CONCLUSIONS AND FUTURE WORK

This paper demonstrates the effectiveness of using informative interfaces to increase the quantity and quality of trainer feedback, using the TAMER framework as a platform for our investigation. Our empirical user study showed that these informative behaviors can significantly increase a trainer's engagement along several different metrics, including the duration of training, the number of feedback instances and the frequency of feedback. Though not significant, the performance-informative behavior led to substantially better agent performance, whereas the uncertainty-informative behavior led to worse agent performance. Further investigation of our experimental data using PCA suggested that this may be because the performance-informative behavior keeps the trainer focused on giving feedback to similar states, whereas the uncertainty-informative behavior induces the trainer to give feedback in a wider range of states. This in turn aligns with the notion "you get what you measure"—measuring performance increased performance, and measuring uncertainty reduced uncertainty.

Future work will focus on developing new interfaces to capitalize on the success of the performance-informative interface, e.g., by creating a tournament and hall of fame with which trainers compete against each other. We will also investigate ways to improve the uncertainty-informative interface by developing different uncertainty metrics and incorporating uncertainty into the agent's learning algorithm. In the longer term, we hope to develop richer interfaces—e.g., using agent avatars and dialogue systems—that are built upon empirical insights from this paper and similar future investigations. We hope that this will create the enduring and effective interaction needed to enable an agent to learn a set of complex and interrelated tasks from human feedback.

8. ACKNOWLEDGMENTS

We thank the anonymous referees for their constructive comments that helped to improve the article. Guangliang Li is supported by China Scholarship Council.

9. REFERENCES

[1] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. *ICML*, 2004.

[2] B. Argall, B. Browning, and M. Veloso. Learning by demonstration with critique from a human teacher. *HRI*, 2007.

[3] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 2009.

[4] D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[5] B. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M. Johnson, and B. Tomlinson. Integrated learning for interactive synthetic characters. *ACM Transactions on Graphics*, 2002.

[6] N. Bohm, G. Kokai, and S. Mandl. Evolving a heuristic function for the game of Tetris. *Proc. Lernen, Wissensentdeckung und Adaptivitat LWA*, 2004.

[7] E. Bouwers, J. Visser, and A. Van Deursen. Getting what you measure. *Communications of the ACM*, 2012.

[8] C. Chao, M. Cakmak, and A. Thomaz. Transparent active learning for robots. *HRI*, 2010.

[9] S. Chernova and M. Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 2009.

[10] E. Demaine, S. Hohenberger, and D. Liben-Nowell. Tetris is hard, even to approximate. *Computing and Combinatorics*, 2003.

[11] D. Gill and T. Deeter. Development of the sport orientation questionnaire. *Research Quarterly for Exercise and Sport*, 1988.

[12] K. Judah, S. Roy, A. Fern, and T. Dietterich. Reinforcement learning via practice and critique advice. *Proc. of the 24th AAAI Conference on AI*, 2010.

[13] W. Knox. *Learning from Human-Generated Reward*. PhD thesis, 2012.

[14] W. Knox, B. Glass, B. Love, W. Maddox, and P. Stone. How humans teach agents. *IJSR*, 2012.

[15] W. Knox and P. Stone. Interactively shaping agents via human reinforcement: The TAMER framework. *Proc. of the 5th International Conference on Knowledge Capture*, 2009.

[16] W. Knox and P. Stone. Combining manual feedback with subsequent MDP reward signals for reinforcement learning. *AAMAS*, 2010.

[17] W. Knox and P. Stone. Reinforcement learning from human reward: Discounting in episodic tasks. *RO-MAN*, 2012.

[18] W. Knox and P. Stone. Reinforcement learning from simultaneous human and MDP reward. *AAMAS*, 2012.

[19] E. Lawrence, P. Shaw, D. Baker, S. Baron-Cohen, A. David, et al. Measuring empathy: reliability and validity of the empathy quotient. *Psychological Medicine*, 2004.

[20] A. Lockerd and C. Breazeal. Tutelage and socially guided robot learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[21] R. Maclin and J. Shavlik. Creating advice-taking reinforcement learners. *Machine Learning*, 1996.

[22] P. Pilarski, M. Dawson, T. Degris, F. Fahimi, J. Carey, and R. Sutton. Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. *International Conference on Rehabilitation Robotics*, 2011.

[23] H. Suay and S. Chernova. Effect of human guidance and state space size on interactive reinforcement learning. *RO-MAN*, 2011.

[24] R. Sutton and A. Barto. *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998.

[25] I. Szita and A. Lorincz. Learning Tetris Using the Noisy Cross-Entropy Method. *Neural Computation*, 2006.

[26] M. Taylor and S. Chernova. Integrating human demonstration and reinforcement learning: Initial results in human-agent transfer. *AAMAS Workshop*, 2010.

[27] A. Tenorio-Gonzalez, E. Morales, and L. Villaseñor-Pineda. Dynamic reward shaping: training a robot by voice. *Advances in Artificial Intelligence–IBERAMIA*, 2010.

[28] A. Thomaz and C. Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. *Proc. of the National Conference on AI*, 2006.

[29] A. Thomaz and C. Breazeal. Transparency and socially guided machine learning. *ICDL*, 2006.

[30] A. Thomaz, G. Hoffman, and C. Breazeal. Real-time interactive reinforcement learning for robots. *AAAI Workshop*, 2005.

[31] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 1992.