4th International Conference on
Development and Learning and on Epigenetic Robotics
October 13-16, 2014. Palazzo Ducale, Genoa, Italy

Learning from Human Reward Benefits from Socio-competitive Feedback

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

Hayley Hung
Technical University of Delft
Delft, Netherlands
h.hung@tudelft.nl

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

W. Bradley Knox
Massachusetts Institute of Technology
Cambridge, USA
bradknox@mit.edu

***Abstract*—Learning from rewards generated by a human trainer observing an agent in action has proven to be a powerful method for non-experts in autonomous agents to teach such agents to perform challenging tasks. Since the efficacy of this approach depends critically on the reward the trainer provides, we consider how the interaction between the trainer and the agent should be designed so as to increase the efficiency of the training process. This paper investigates the influence of the agent's *socio-competitive feedback* on the human trainer's training behavior and the agent's learning. The results of our user study with 85 subjects suggest that the agent's socio-competitive feedback substantially increases the engagement of the participants in the game task and improves the agents' performance, even though the participants do not directly play the game but instead train the agent to do so. Moreover, making this feedback active further induces more subjects to train the agents longer but does not further improve agent performance. Our analysis suggests that this may be because some trainers train a more complex behavior in the agent that is appropriate for a different performance metric that is sometimes associated with the target task.**

I. Introduction

Autonomous agents have the potential to play a transformative role in many aspects of society in the near future. However, to realize this potential, such agents need to be able to efficiently learn how to perform challenging tasks from humans who, although experts in the tasks they are teaching, may have little expertise in autonomous agents or computer programming. Therefore, there is a great need for new methods that facilitate the interaction between humans and agents through which such learning occurs.

The feedback that the human provides during such interaction can take many forms, e.g., reward and punishment [1], [2], [3], advice [4], guidance [5], or critiques [6]. Within them, learning from rewards generated by a human trainer observing the agent in action has proven to be a powerful method for human trainers who are not experts in autonomous agents to teach such agents to perform challenging tasks. However, how to make an agent learn efficiently from these kinds of human rewards is still under-addressed. Since the efficacy of this approach depends critically on the reward the trainer provides, we consider how the interaction between the trainer and the agent should be designed so as to increase the efficiency of this learning process.

In earlier research, we showed that the way that the agent interacts with the human trainer can greatly affect the trainer's engagement and agent's learning. In particular, we showed that if an agent keeps the trainer informed about the agent's past and current performance, the trainer will provide more feedback and the agent will ultimately perform better [7]. Hence, this result shows that the interaction between the agent and the trainer should ideally be *bi-directional*: not only should the trainer give the agent the feedback it needs for learning, the agent should explicitly give the trainer feedback on how well that learning is going.

In this paper, we seek to build on this work by investigating how to improve the sophistication and efficacy of such a bi-directional interface. In particular, we propose a new *Socio-competitive TAMER* interface, in which the trainer is embedded in an environment that makes her aware of other trainers and their respective agents. To this end, we developed a new Facebook app that implements such a social interface. In addition to receiving feedback about how her agent is performing, the trainer now also sees a leaderboard that compares her agent's performance to that of her Facebook friends as well as all others using the Facebook app. We hypothesize that putting the trainers in an environment in which they compete with each other can further motivate them to provide more and better feedback to their agents.

In addition, we propose a second extension in which the agent *actively* provides feedback to the trainer. While both the interface in [7] and the social extension mentioned above are bi-directional, the agent's role is passive: it merely displays feedback for the trainer, which the trainer can choose to look at or ignore. To address this limitation, we developed an extension to the Facebook app that uses Facebook *notifications*, i.e., messages sent to Facebook users while they are not using the app, that update the trainers on their performance relative to other trainers. We hypothesize that actively providing the trainer with feedback in this way will motivate trainers to return more often to the training process, resulting in more feedback for the agent and better ultimate performance.

To test these hypotheses, we conducted an experiment with 85 subjects applying our Socio-competitive TAMER interface to the game of Tetris. The results of our user study with 85 subjects suggest that the agent's socio-competitive feedback substantially increases the engagement of the participants in the game task and improves the agents' performance, even though the participants do not directly play the game but instead train the agent to do so. Moreover, making this feedback active further induces more subjects to train the agents longer but does not further improve agent performance. A deeper analysis suggests that some of these trainers were training more complex agent behavior by optimizing on their own performance metric.

The rest of this paper begins with a review of related work in Section II and provides background on TAMER in Section III. Section IV introduces the proposed Socio-competitive TAMER interface and Section V presents the experimental conditions. Section VI describes the experimental setup, and Section VII reports and discusses the results. Finally, Section VIII concludes and discusses future work.

II. Related Work

In this section, we discuss related work in learning from human rewards, social networks, and gamification.

A. Learning from Human Rewards

An agent can learn from human feedback about the agent's behavior. In this learning scenario, feedback can be restricted to express various intensities of approval and disapproval; such feedback is mapped to numeric "reward" that the agent uses to revise its behavior [2], [3], [8], [1], [9]. Compared to learning from demonstration, learning from human reward requires only a simple task-independent interface and may require less expertise and place less cognitive load on the trainer [10].

The TAMER framework [3] allows an agent to learn from human reward signals instead of environmental rewards. Using TAMER as a foundation, Knox et al. [11] examine how human trainers respond to changes in their perception of the agent and to certain changes in the agent's behavior, while Li et al. [7] investigate how informative feedback from the agent affects trainers' behaviors. Knox et al. find that the agent can induce the human trainer to give more feedback but with lower performance when the quality of the agent's behavior is deliberately reduced whenever the rate of human feedback decreases. Li et al. show that more and higher quality feedback is elicited from the trainers when the agent's past and present performance is displayed to the trainer.

The approach of Knox et al. investigates how an agent's task-focused behavior affects a trainer's training behavior. However, the approach of Li et al. suggests that the agent should also provide information about its learning process to the trainer. Ultimately, we believe that it will be helpful for facilitating the interaction between the trainer and the agent if the agent provides information (such as facial expressions, body language, and gaze behavior) to indicate something about its learning state and solicit feedback from a human [12], [13], [14]. However, as an early step towards this goal, we concentrate in this work on analyzing how sharing socially derived competitive information can influence the trainer's behavior.

B. Social Networks

Research on Online Social Networks (OSNs) emanates from a wide variety of disciplines and involves research such as descriptive analysis of users, motivations for using Facebook, identity presentation, the role of Facebook in social interactions, and privacy and information disclosure [15]. Some researchers use OSNs as a tool for recruiting subjects and testing hypotheses. Many OSNs such as Facebook and MySpace have opened themselves to developers, enabling them to create applications that leverage their users' social graphs. For example, Nazir et al. [16] created three popular applications with Facebook Developer, a platform for developers to build social apps on Facebook. These apps had over eight million users, providing an enormous data set for research. Through their social game Magpies, also on Facebook, Kirman et al. [17] found that the additional socio-contextual data such as social network analysis (SNA) information can increase the frequency of social activity between players engaged in the game but does little to increase the growth of the player-base. Similar to this paper, Rafelsberger and Scharl [18] propose an application framework to develop interactive games with a purpose on top of social networking platforms, leveraging the wisdom of the crowd by engaging users in online games to complete tasks that are trivial for humans but difficult for computers. In this paper, we leverage the social aspects of OSNs both to recruit human trainers and to improve the performance of the agents they train.

C. Gamification

Gamification is defined as the use of game-design elements (such as a score or leaderboard) in non-game contexts [19]. Recently researchers and practitioners in the field of online marketing, digital marketing and interaction design have begun to apply gamification to drive user engagement in non-game application areas including productivity, finance, health, education and sustainability [20], [21].

For instance, Dominguez et al. [22] used gamification as a tool to increase student engagement by building a gamification plugin for an e-learning platform. They demonstrated that students who completed the gamified experience got better scores in practical assignments (such as how to complete different tasks using a given application, e.g. word, spreadsheet), but performed poorly on written assignments and participated less in class activities.

Inspired by these works, in this paper, we incorporate gamification into agent training by embedding the game in an OSN with competitive elements, aiming to increase the amount of time spent and feedback given by a trainer to further improve agent performance.

III. Background

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

A. TAMER Framework

An agent implemented according to the TAMER framework learns from real-time evaluations of its behavior, provided by a human trainer. From these evaluations, which we refer to as “reward”, the TAMER agent creates a predictive model of future human reward and chooses actions it predicts will elicit the greatest human reward. Unlike in traditional reinforcement learning, a reward function is not predefined.

A TAMER agent strives to maximize the reward caused by its immediate action, which also contrasts with traditional reinforcement learning, in which the agent seeks the largest discounted sum of future rewards. The intuition for why an agent *can* learn to perform tasks using such a 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. Until recently [23], general myopia was a feature of all algorithms involving learning from human feedback and has received empirical support [10]. Built to solve a variant of a Markov decision process, (i.e., a specification of a sequential decision-making problem commonly addressed through reinforcement learning [24]) in which there is no reward function encoded before learning, the TAMER 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 can give reward corresponding to its quality.

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 sum of reward instances observed shortly after taking action a in state s.

In TAMER, 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 (upto ± 4). The TAMER learning algorithm repeatedly takes an action, senses reward, and updates $\hat{H}$.

B. Tetris Platform

Tetris is a fun and popular game that is familiar to most people, making it an excellent platform for investigating how humans and agents interact during agent learning. We use an adaptation of the RL-Library implementation of Tetris.[1]

Although Tetris 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 [25]. 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 previously described 23 features [26]—to represent the state observation. The input to $\hat{H}$ is 46 corresponding state-action features, the difference between state features before a placement and after the placement and clearing any resulting solid rows.

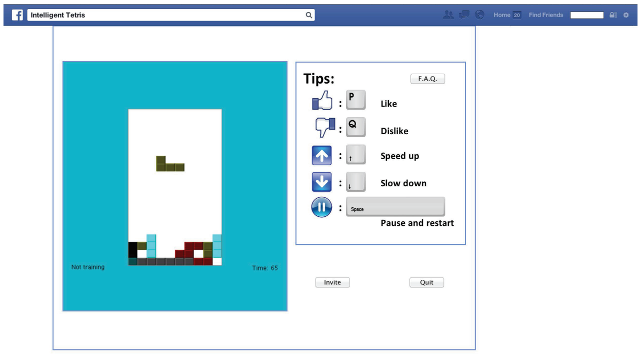


Fig. 1: Intelligent Tetris, our Facebook app for training.

Like other implementations of Tetris learning agents (e.g., [27], [28], [29]), 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.

IV. Socio-competitive TAMER Interface

Our Socio-competitive TAMER interface was developed by integrating the original TAMER interface into a Facebook frame with Facebook Developer. To our knowledge, this interface is the first to incorporate TAMER into a social-network setting. The interface eases subject recruitment for the experiment by leveraging a subject’s social network to gain more participants. Moreover, the interface enables the experiment to be integrated into people’s daily lives, and thereby gather data in a realistic context.

The Socio-competitive TAMER interface facilitates the development of social apps for numerous different agent training tasks with TAMER in a social network setting. For this paper, we developed the Facebook App ‘Intelligent Tetris’ as a platform for our experiments. The Facebook user can visit the app via a Facebook page describing the experiment, or by searching for it in the App Center. By clicking on the “Play game” button, the user can enter into the app page. To start training, the user must agree with the permissions, terms and conditions to authenticate the app. As shown in Figure 1, the training page contains a game board on the left side and a tip box on the right that shows training instructions.

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

TABLE I: Summary of the four conditions. As described in Section V, the non-social feature is a display of the agent's performance history, the passive social feature is the leaderboard, and the active social feature is the notifications of changes in a user's leaderboard rank. Note that all conditions allowed people to invite their friends to install the app, thereby becoming subjects.

Condition	Non-social behavior	Social behavior	
		Passive	Active
Control			
Performance	✓		
Passive Social	✓	✓	
Active Social	✓	✓	✓

The key advantage of the Socio-competitive TAMER interface is that, as with other experimental uses of Facebook [16], [17], [18], many users can be recruited in a short time, making research in this area more feasible. In our experiment, 100 subjects consented to install the app within the first three days of this study. By contrast, our earlier experiment [7] obtained only 51 subjects using a more aggressive recruitment effort that included manually sending emails to potential subjects, putting up flyers and posters, and sending reminder emails.

V. Experimental Conditions

In this section, we present the four conditions used in our experiment. The control and performance conditions are replicated from our earlier work [7]. The passive and active social conditions are the novel conditions we propose in this paper. The conditions and their corresponding functionalities are summarized in Table I.

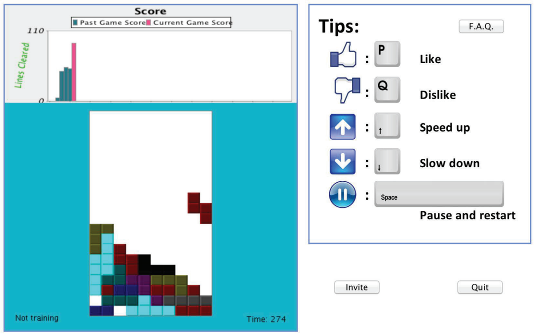


Fig. 2: The performance interface.

A. Control Condition

The interface for the *control* condition is the original TAMER interface presented in [3] but placed within the Socio-competitive TAMER interface, as described in Section IV. The trainer is not given any feedback except for the state and action of the agent, which are visible from the Tetris game board. Participants can give positive and negative feedback to the previous action of the agent. They can increase the strength of this feedback by pressing the button more times (up to a range of ± 4).

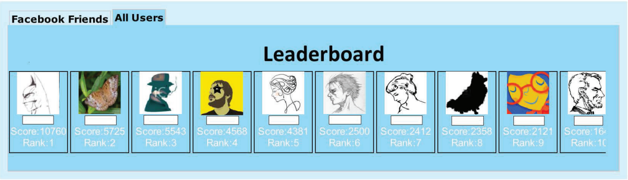


Fig. 3: The leaderboards. (The profile images and names are obscured for anonymization reasons.)

B. Performance Condition

The *performance* condition is implemented by integrating the performance-informative interface of [7] into the Socio-competitive TAMER interface. Here, the agent's performance over past and current games is shown in a performance window during the training process. As shown in Figure 2, each bar in the performance window indicates the agent's performance in one game chronologically from left to right. The agent's performance is measured by the number of lines cleared. During training, the pink bar represents the number of lines cleared so far for the current game, while the dark blue bars represent the performance of past games. When a game ends, the corresponding bar becomes dark blue and any new lines cleared in the new game are visualized by a pink bar to its right. When the performance window is full, the window is cleared and new bars appear from the left.

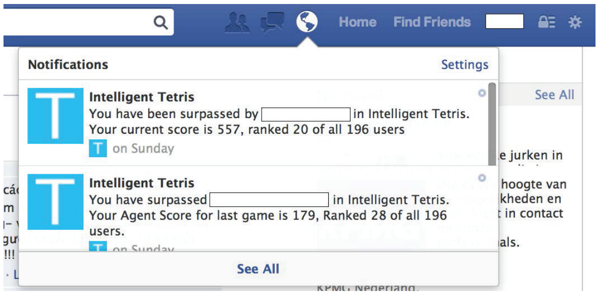
Our earlier work [7] found that this performance-informative interface, in comparison to the control interface, can increase the duration of training, the amount of feedback from the trainer, and the agent's performance.

C. Passive Social Condition

In the *performance* condition, the agent shows only its own performance to the human trainer. We hypothesize that people will be further motivated to improve the agent's performance if they are put in a socio-competitive situation where they can compare the performance of their agents with that of others. Therefore, in the *passive social* condition, we allow the agent to indicate the rank and score of the trainer's Facebook friends, as well as those of all trainers. This condition is called passive because the agent does not actively seek the attention of the trainer. This information is also displayed only within game play, unlike the active social condition discussed below.

To implement this condition, we added a leaderboard on top of the interface of the *performance* condition. An example leaderboard is shown in Figure 3. There are two leaderboards in the leaderboard frame: 'Facebook Friends' and 'All Users'. For each trainer, all her friends who registered the app are listed in the 'Facebook Friends' leaderboard and all the participants who registered the app are listed in the 'All Users' leaderboard. The trainer's Facebook friends and other participants in the leaderboard can also be in other conditions. The first name and profile image of each trainer from her respective Facebook account is shown in the leaderboards.

When the trainer starts training for the first time, her agent's performance is initialized to 0 and ranked in the



Intelligent Tetris
You have been surpassed by □ in Intelligent Tetris.
Your current score is 557, ranked 20 of all 196 users
on Sunday

Intelligent Tetris
You have surpassed □ in Intelligent Tetris.
Your Agent Score for last game is 179, Ranked 28 of all 196 users.
on Sunday

Fig. 4: Notification (with names anonymized).

leaderboard. Whenever the trainer finishes a game, the new game score and rank is updated in the two leaderboards. To create more movement up and down the leaderboard, only the latest game score is used. The trainer can check her score and rank in each leaderboard by moving the cursor over the corresponding tab. Even when the trainer quits training without finishing the game, the game is finished for her offline and a new game score and rank is updated to both leaderboards. Therefore, the trainer can keep track of both the agent's learning progress *and* the agent's performance relative to that of her friends and all other trainers.

D. Active Social Condition

In the *performance* and *passive social* conditions, the performance information is only passively shown by the agent within game play. Intuitively, as between human teachers and students, the interaction between the human trainer and agent should not only be bi-directional, but both the student and teacher should take active roles. Therefore, in the *active social* condition, we allow the agent to notify the human trainer *outside* of the socio-competitive TAMER app about its performance *relative* to others. We hypothesize that actively informing the trainer in this way will encourage the trainer to return to the application and further motivate her to improve the agent's performance.

In this condition, in addition to the leaderboards, at the end of each training session, when the user in this condition quits training without finishing the last game, the app finishes the game offline. A notification is sent to the trainer. On Facebook, app notifications are short free-form messages of text. They can effectively communicate important events, invites from friends or actions people need to take. When a notification is delivered, it highlights the notifications jewel on Facebook and appears in a drop-down box when clicked. An app notification is displayed to the right of the corresponding app's icon, interspersed with other notifications in chronological order, as shown in Figure 4. Note that this notification is not actively shown in a pop-up display. The trainer can only see the contents of the notification by clicking on the highlighted notification jewel.

In this condition, notifications about the agent's performance are sent to the user if the rank of her agent has increased or decreased relative to others. Likewise, if another agent surpasses, or is just surpassed by the current trainer's agent, those corresponding trainers in the *active social* condition are also notified of the change in their agents' ranks. Note that if an agent jumps several ranks, only the nearest ranked agent to the new location is considered for the notification. To ensure that the leaderboards were well-populated, the ranks of all subjects were used.

More precisely, for the user whose new game score surpasses others, a notification saying 'You have surpassed ___ in Intelligent Tetris. Your agent score for last game is ___, ranked ___ of all ___ users.' is sent to the current user; for the user being surpassed, she receives a notification saying 'You have been surpassed by ___ in Intelligent Tetris. Your current game score is ___, ranked ___ of all ___ users.', as shown in Figure 4.

There are many other ways the agent could notify the trainer, e.g., by posting on the user's wall or newsfeed. However, we were concerned these approaches would carry a higher risk of annoying the trainer. In addition, the resulting information would be seen by the trainer's friends or other trainers who are in other conditions, creating a confounding factor in our experiment. Therefore, we only use notifications to implement the active aspect of this condition. To avoid annoying the trainer, at most three notifications are sent every 24 hours.

VI. Experimental Setup

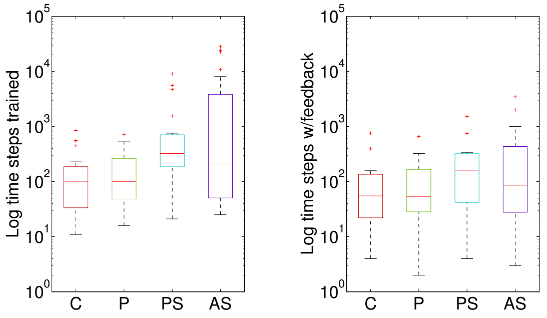
To evaluate our interfaces, we conducted an experiment with our 'Intelligent Tetris' Facebook App. 157 participants were recruited and uniformly distributed into the four conditions. However, eight participants started training but never gave any feedback and 64 participants registered the app but did not start training. Therefore, only data from the remaining 85 participants (69 male and 15 female) were analyzed. Of these, 66 were from Europe, 3 from North America, 3 from South America, 1 from Asia and 1 from New Zealand, aged from 17 to 46.[2]

There were 21 participants in the *control* condition, 19 in the *performance* condition, 20 in the *passive social* condition and 25 in the *active social* condition. The experiment started on June 18, 2013 and ended on July 18, 2013. For each trainer, we recorded state observations, actions, human rewards, lines cleared, timestamp of mouse-overs of the leaderboard tabs, content and timestamp of notifications, as well as other user information such as email address, location, age, gender, etc. There was a FAQ page displaying the instructions for training and problems that may occur when registering the app. The user could also visit a separate Facebook page dedicated to the experiment via a link in the FAQ page where detailed terms and conditions regarding consent were provided. Unlike our earlier experiment [7], the trainers were not given time to practice before training started.

[2] Note that not all participants provided demographic information via their Facebook accounts.

VII. Results and Discussion

We present and analyze the results of our experiment in this section. In the results below, the p-value was computed with the non-parametric Mann-Whitney-Wilcoxon test (one-tailed). In the box plots, the bottom and top of the box are the first and third quartiles, and the line inside the box is the second quartile (the median). The range spanned by the box is the interquartile range (IQR). The plotted whiskers extend to the most extreme data value that is not an outlier; data values are considered outliers (and drawn as plusses) if they are 1.5 × IQR larger than the third quartile or 1.5 × IQR smaller than the first quartile.



(a) (b)

Fig. 5: Boxplots across the four conditions of (a) total time steps trained by subjects and (b) between-subject distribution of the total number of time steps that were labeled with feedback. C: control; P: performance; PS: passive social; AS: active social.

A. Training Time

Figure 5(a) summarizes the total number of time steps trained for each condition (note the log scale). A time step equates to the execution of one action by the agent, which is a metric unaffected by the trainer's chosen falling speed. The results show that, in the passive social condition, the subjects trained significantly longer than in either the control (3.3 times in median; $U = 114.5$, $z = -2.48$, $p < 0.01$, $r = 0.39$) or performance conditions (3.2 times in median; $U = 110.5$, $z = -2.22$, $p < 0.02$, $r = 0.36$). Similarly, in the active social condition, the subjects also trained longer than in either the control (2.2 times in median; $U = 188$, $z = -1.63$, $p = 0.05$, $r = 0.24$) or performance conditions (2.2 times in median; $U = 179$, $z = -1.37$, $p = 0.08$, $r = 0.21$). In addition, in the active social condition, the subjects trained less in median than in the passive social condition (0.7 times). However, the active social condition resulted in longer mean training time than the passive social condition (3.4 times). Thus, the social conditions positively affected training time, which is consistent with our hypotheses.

B. Amount of feedback

Figure 5(b) summarizes the distribution of number of time steps with feedback for all the subjects in the four conditions. The results show that, in the passive social condition, the trainers gave more feedback than in either the control (2.9 times in median; $U = 152.5$, $z = -1.49$, $p = 0.07$, $r = 0.23$) or performance conditions (3.0 times in median; $U = 143.5$, $z = -1.29$, $p = 0.098$, $r = 0.21$). Similarly, in the active social condition, the trainers also gave more feedback than in either the control (1.6 times in median; $U = 202$, $z = -1.32$, $p = 0.09$, $r = 0.20$) or performance conditions (1.6 times in median; $U = 199$, $z = -0.90$, $p = 0.18$, $r = 0.14$). In addition, in the active social condition the trainers gave less feedback in median than in the passive social condition (0.6 times), but the active social condition resulted in more mean time steps with feedback given than the passive social condition (1.6 times). Again, consistent with our hypotheses, the social conditions positively affected the quantity of time steps with feedback.

We also analyzed the total instances of feedback, where an instance is a single press of the feedback button and there can be multiple presses for one time step. In the passive social condition, the trainers gave more feedback instances (number of times positive or negative feedback button was pressed) than in either the control (3.2 times in median; $U = 136.5$, $z = -1.90$, $p < 0.05$, $r = 0.30$) or performance conditions (2.2 times in median; $U = 146$, $z = -1.22$, $p = 0.11$, $r = 0.20$). Similarly, in the active social condition, the trainers also gave more feedback instances than in either the control (2.0 times in median; $U = 194.5$, $z = -1.49$, $p = 0.07$, $r = 0.22$) or performance conditions (1.4 times in median; $U = 198.5$, $z = -0.91$, $p = 0.18$, $r = 0.14$). However, in the active social condition the trainers gave less feedback in median than in the passive social condition (0.6 times), but the active social condition resulted in slightly more mean feedback instances than the passive social condition (1.1 times).

Overall, our results suggest that the agents' social performance feedback can influence the trainer to give more feedback and spend more time on training.

C. Performance

We hypothesized that the trainer's increased engagement (i.e., not only more training time, but also motivation to give more and better feedback) would lead to improved performance by the agents. To test this, we first examined how the agents' performances varied as the trained policy changed over time. We divided up the training time of each trainer into intervals. The first six intervals consist of 50 time steps each, next two of 100 time steps each, and thereafter, all intervals consist of 200 time steps each.

For each subject, the agent's policy was saved at the end of each interval and tested offline for 20 games, since the states visited for each game can vary a lot. However, some factors such as the distribution of the trainer's skill level across conditions, the domain stochasticity etc., may still affect the evaluation of agent performance. Nonetheless, we believe that the large number of participants can compensate for these variabilities encountered while running studies in the wild. The performance for each condition was computed by averaging across the 20 offline games, then across all

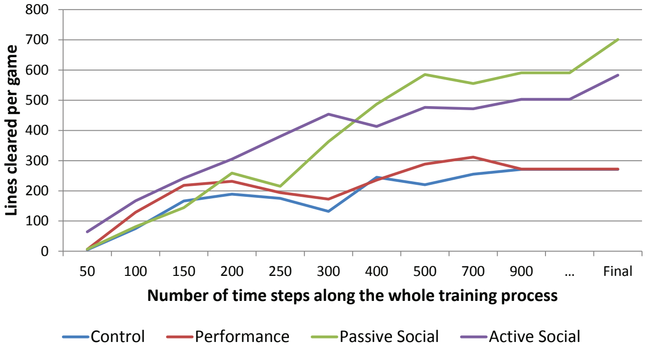


Fig. 6: Mean offline performance.

the subjects in each condition. If the subject's final training instances stop sooner than the final interval, the performance of her agent's final policy is taken in later intervals. The longest-trained agents in the control and performance conditions received up to 10 intervals of feedback, whereas for the passive and active social conditions the longest-trained agents received 51 and 146 intervals of feedback respectively. Like the performance measure in [7], in the analysis below we use the mean value across subjects as the performance for each condition.

As shown in Figure 6, early performance within the passive social condition was similar to that within the control condition, whereas the performance within both the active social and performance conditions was better. Thereafter, performance within the passive social condition increases faster than the other three conditions, and ultimately becomes higher than that of the other three conditions. The active social condition, which performs the second best overall, outperforms the control condition throughout the training process and outperforms the performance condition from 200 time steps on.

Thus, these results suggest that the social conditions improved the performance of the agent. Surprisingly, however, and not consistent with our hypothesis, the active social condition did not significantly outperform the passive social condition ($p = 0.22$). From the results in Section VII-A we see that trainers in the active social condition, trained slightly less in median (0.7 times) than in the passive social condition but much longer in mean (3.4 times), indicating that more trainers come back to further train agents because of the active social behavior (i.e. receiving notifications) but did not tend to achieve better agent performance. As a possible explanation, we considered whether some trainers employed a strategy of training the agent to clear multiple lines at once, which is given more points per line in some traditional Tetris games, but not in our experiment. Since such a strategy is more complex, it could take longer to train, and its effects might therefore be greater in the *active social* condition, where trainers did the most training.

To investigate this further, we retested the final offline performance with the score mechanism for the original Tetris game, hypothesizing that some trainers were employing a multi-line strategy and that trainers in the *active social* condition, by persisting longer with this more complex strategy, would benefit the most from a score mechanism that rewards clearing multiple lines at once. This score mechanism gives increasing bonuses (0.5, 4.5, 26) for clearing 2, 3, or 4 lines respectively. We found that the new offline performance for the *active social* condition (719 lines) increased more than the new offline performance of the *passive social* condition (728 lines), when compared to their corresponding bonus-free performances (583 and 700 lines, respectively).

This difference in training behavior between the two conditions was not expected and further analysis is required to understand why the trainers in the *active social* condition stopped training before their agent performance surpassed those of the *passive social* condition. One possible explanation is that the *active social* trainers stopped training because their rankings were not improving even with their more complex training strategy. However, we do not know whether this multi-line strategy was trained during the first interval or emerged later on in the training process. We suspect the reason that the rankings did not improve is the large variance of agent performance in Tetris, which could frustrate the trainer when the agent's online performance is low despite continued training and a good learned policy.

D. Influence of Social Information

a) Looking at the Leaderboards: To measure the extent to which social information influenced the trainers, we tried to measure how often they looked at the leaderboard. Since we cannot measure this directly, we used the number of mouseovers as a proxy for this. Our data shows that more than half of the participants in the *passive* and *active* social conditions moved the cursor over the leaderboard tabs at least once. In the passive social condition, 11 of 19 trainers moused over the leaderboard tabs, where 5 of them moused over more than 10 times and one even checked up to 31 times in five days. In the active social condition, 17 of 25 trainers moused over the leaderboard tabs, where 5 of them moused over more than 10 times and one did this up to 40 times. Using Pearson's correlation test, we also observed that for both conditions, the number of tab mouseovers correlates with the number of time steps trained ($r = 0.60, p \approx 0.006$ and $r = 0.89, p \approx 0$ for passive social and active social conditions respectively) and the trained agents' final offline performances ($r = 0.72, p \approx 0.0004$ and $r = 0.67, p \approx 0.0002$ for passive social and active social conditions respectively).

b) Receiving Notifications: The data shows that 22 trainers received 40 notifications in total in the *active social* condition. 6 of the 22 trainers received 9 notifications saying she had been surpassed by others, and the other 31 notifications informed the trainer she surpassed other trainers. The notification jewel was clicked 28 times by 11 of the 22 trainers, where 8 of them clicked more than once. Pearson's correlation test shows that the number of notifications the trainer received correlates with the time steps trained ($r = 0.18, p = 0.39$) and the number of time steps with feedback ($r = 0.41, p = 0.04$).

VIII. Conclusion and Future Work

By integrating agent training with an online social network via the Socio-competitive TAMER interface, this paper investigated the influence of social feedback on human training and the resulting agent performance. With this interface, we addressed the challenge of recruiting subjects and inserted agent training into people's daily online social lives. The results of our user study showed that the agent's social feedback can induce the trainer—possibly by inducing between-trainer competitiveness—to train longer and give more feedback. The agent performance was much better when social-competitive feedback was provided.

We found that adding active social feedback induced more trainers to train longer (on average, but less in median) and provide more feedback but did not further improve agent performance. A deeper analysis of the behaviors of the trainers is needed to understand why more subjects were induced by the active social behavior to train longer and give more feedback while still not surpassing the performance of agents in the passive social condition. One possible explanation is that some trainers employed a more complex multi-line clearing strategy, which had greater effects for the active social condition, where the most training occurred, or it could be related to the median training time being slightly lower. This is tentatively suggested by the substantial increase in performance of the agents in this condition when using a scoring system that rewards clearing multiple lines rather than single line. In contrast, in the passive social condition, the increase in performance when using the multi-line score weighting is much less. We suspect that the reason for not further training the agent to successfully learn a multi-line clearing strategy can be the huge variance of agent performance in Tetris. Finally, we believe that our approach could transfer to other domains and methods for agent learning from a human, since TAMER succeeds in many domains including Tetris, Mountain Car, Cart Pole, Keepaway Soccer, Interactive Robot Navigation etc. [26], though more investigations are needed to test this hypothesis.

Future work will focus on further investigating how trainers' relationships with each other impact their training. For example, people with closer relationships may be more competitive with each other than with strangers. In addition, we would also like to investigate how multiple trainers can train a single agent together and whether the social dynamic between trainers could also positively influence training in such a scenario.

Acknowledgments

We thank the anonymous referees for their constructive comments. Guangliang Li is supported by the China Scholarship Council.

References

[1] 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.

[2] C. Isbell, C. Shelton, M. Kearns, S. Singh, and P. Stone, "A social reinforcement learning agent," *Proc. of the 5th International Conference on Autonomous Agents*, 2001.

[3] W. Knox and P. Stone, "Interactively shaping agents via human reinforcement: The TAMER framework," *International Conference on Knowledge Capture*, 2009.

[4] R. Maclin, J. Shavlik, L. Torrey, T. Walker, and E. Wild, "Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression," in *National Conference on Artificial Intelligence*. AAAI Press, 2005.

[5] 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.

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

[7] G. Li, H. Hung, S. Whiteson, and W. B. Knox, "Using informative behavior to increase engagement in the tamer framework," *AAMAS*, 2013.

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

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

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

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

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

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

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

[15] R. E. Wilson, D. G. Samuel, and L. T. Graham, "A review of facebook research in the social sciences," *Perspectives on Psychological Science*, 2012.

[16] A. Nazir, S. Raza, and C.-N. Chuah, "Unveiling facebook: a measurement study of social network based applications," in *ACM SIGCOMM*, 2008.

[17] B. Kirman, S. Lawson, C. Linehan, F. Martino, L. Gamberini, and A. Gaggioli, "Improving social game engagement on facebook through enhanced socio-contextual information," in *SIGCHI*, 2010.

[18] W. Rafelsberger and A. Scharl, "Games with a purpose for social networking platforms," in *ACM Conference on Hypertext and Hypermedia*, 2009.

[19] S. Deterding, R. Khaled, L. E. Nacke, and D. Dixon, "Gamification: Toward a definition," in *CHI 2011 Gamification Workshop*, 2011.

[20] G. Zichermann and J. Linder, *Game-based marketing: inspire customer loyalty through rewards, challenges, and contests*. Wiley. com, 2010.

[21] K. M. Kapp, *The gamification of learning and instruction: game-based methods and strategies for training and education*. Wiley. com, 2012.

[22] A. Domínguez, J. Saenz-de Navarrete, L. De-Marcos, L. Fernández-Sanz, C. Pagés, and J.-J. Martínez-Herráiz, "Gamifying learning experiences: Practical implications and outcomes," *Computers & Education*, 2013.

[23] W. B. Knox and P. Stone, "Learning non-myopically from human-generated reward," in *IUI*, 2013.

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

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

[26] W. Knox, "Learning from human-generated reward," Ph.D. dissertation, University of Texas at Austin, 2012.

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

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

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