Social interaction for efficient agent learning from human reward

Li, Guangliang; Whiteson, Shimon; Bradley Knox, W; Hung, Hayley

DOI
10.1007/s10458-017-9374-8

Publication date
2018

Document Version
Final published version

Published in
Autonomous Agents and Multi-Agent Systems

Auton Agent Multi-Agent Syst (2018) 32:1–25
https://doi.org/10.1007/s10458-017-9374-8

Social interaction for efficient agent learning from human reward

Guangliang Li[1] · **Shimon Whiteson**[2] · **W. Bradley Knox**[3] · **Hayley Hung**[4]

Published online: 3 July 2017

Abstract Learning from rewards generated by a human trainer observing an agent in action has been proven to be a powerful method for teaching autonomous agents to perform challenging tasks, especially for those non-technical users. 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 article 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 participants suggest that the agent's passive socio-competitive feedback—showing performance and score of agents trained by trainers in a leaderboard—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.

This article is an extension of our earlier work in [27,28]. It provides a more extensive review of related work on learning from human reward and detailed discussion over gamification in motivating the gamification techniques we used in our system and highlighting the novelty of our approach in the context of gamification, and significantly extends upon our initial work by providing a more extensive analysis of the effect of agent's social competitive feedback on the human trainer's training behavior and agent learning performance and the real effect of the additional active socio-competitive feedback.

✉ Guangliang Li
guangliangli@ouc.edu.cn

Shimon Whiteson
shimon.whiteson@cs.ox.ac.uk

W. Bradley Knox
bradknox@mit.edu

Hayley Hung
h.hung@tudelft.nl

1 Ocean University of China, Qingdao, China

2 University of Oxford, Oxford, UK

3 Massachusetts Institute of Technology, Cambridge, MA, USA

4 Delft University of Technology, Delft, The Netherlands

Moreover, making this feedback active—sending the trainer her agent's performance relative to others—further induces more participants to train agents longer and improves the agent's learning. Our further analysis shows that agents trained by trainers affected by both the passive and active social feedback could obtain a higher performance under a score mechanism that could be optimized from the trainer's perspective and the agent's additional active social feedback can keep participants to further train agents to learn policies that can obtain a higher performance under such a score mechanism.

Keywords Reinforcement learning · Human agent interaction · Learning from human reward · Gamification

1 Introduction

In the future, autonomous agents will operate in human inhabited environments in most real world applications and potentially become an integral part of humans' daily lives. When autonomous agents enter into the real world, they need to adapt to many novel, dynamic and complex situations that cannot be imagined and pre-programmed in the lab and try to learn new skills. Meanwhile, human users may also want to teach agents behaviors they like, most of whom 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 an interaction can take many forms, e.g., reward and punishment [15,21,34], advice [31], guidance [41], or critiques [1]. Within them, interactive shaping is a teaching method that has been developed and proven to be a powerful technique for facilitating autonomous agents to learn how to perform tasks according to a human trainer's preference. In interactive shaping, an agent learns from real time human reward signals, i.e., evaluations of the quality of an agent's behavior delivered by a human observer. However, though the agent can already learn from such human-delivered reward signals, the agent learning critically depends on the quality and quantity of the interaction between the human teacher and the agent. Therefore, in this article, we use interactive shaping, the TAMER framework [21] in particular, and 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 [26,29]. In particular, 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 [29]. 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 article, 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 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 [29] 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 participants applying our Socio-competitive TAMER interface to the game of Tetris. The results of our user study with 85 participants 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 participants to train the agents longer and improves the agent's learning.

Our further analysis suggests that agents trained by trainers affected by both passive and active social feedback learned policies that can get a higher performance under a traditional Tetris game score mechanism that could be optimized from the trainer's perspective, compared to just counting the number of lines cleared. In addition, the effect on performance with such a mechanism started from the beginning of training and increased along the training process. However, this effect plateaued on participants affected by the agent's passive social feedback, while the agent's additional active social feedback can further keep some participants training agents to get a higher performance under such a mechanism.

The rest of this article begins with a review of related work in Sect. 2 and provides background on TAMER in Sect. 3. Section 4 introduces the proposed Socio-competitive TAMER interface and Sect. 5 presents the experimental conditions. Section 6 describes the experimental setup, and Sect. 7 reports and discusses the results. Finally, Sect. 8 concludes.

2 Related work

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

2.1 Learning from human reward

An agent learning from rewards provided by a human is also called "interactive shaping". Rewards provided by a human teacher is termed "human reward", different from the environmental reward provided through a pre-defined reward function in traditional reinforcement learning (RL). Human reward evaluating the quality of an agent's behavior is used as feedback by the human teacher to improve the agent's behavior. This kind of feedback can be restricted to express various intensities of approval and disapproval and mapped to a numeric "reward" for the agent to revise its behavior [15,21,34,36,39]. In contrast to learning from demonstration [2], learning from human reward requires only a simple task-independent interface, potentially less expertise and cognitive load—the amount of mental effort used to teach the agent—from the trainer [23].

Clicker training [4,17] is a related concept that involves using only positive reward to train an agent. It is 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 the desired behavior. In the first work using both reward and punishment to train an artificial agent [15,16], a software agent called Cobot is developed by applying reinforcement learning in an online text-based virtual world where users interact with each other. The agent learned to take proactive actions from multiple sources of human reward, which are 'reward and punish' text-verbs invoked by multiple users.

In addition, Thomaz and Breazeal [41,42] implemented an interface with a tabular *Q-learning* [43] agent which learns an action value function by storing the action value for each state-action pair in a table. A separate interaction channel is provided in the interface allowing the human to give the agent feedback. The agent aims to maximize its total discounted reward, which is the sum of the human reward and environmental reward. They treat the human's feedback as an additional reward that supplements the environmental reward. Moreover, an improvement in the agent performance is shown by allowing the trainer to give action advice on top of the human reward. Suay and Chernova [36] extended their work to a real-world robotic system using only the human reward.

Based on Thomaz and Breazeal's algorithm, Tenorio-Gonzalez et al. [39] proposed an algorithm that learns from both explicit demonstrations and human rewards as well as the environmental reward. The demonstration is first provided via verbal commands and directly used by the robot to initialize the value function with the environmental reward. Then, as in the work of Thomaz and Breazeal, the human and environmental reward signals are added together as a single signal for the robot learning. However, different from Thomaz and Breazeal's work, all human inputs (including demonstrated actions and human rewards) in Tenorio et al.'s algorithm are given via verbal commands or feedback, which could introduce errors since the speech recognition system could misinterpret them. They assume the wrong verbal reward is not given all the time and could be corrected with additional feedback from the user. Their results show that the robot learning with some noisy verbal feedback is slower than with perfect feedback but still faster than learning with traditional RL.

Instead of adding the human reward and environmental reward together as a single signal, Knox and Stone [21] proposed the *TAMER* framework that allows an agent to learn from only human reward signals instead of environmental rewards by directly modeling the human reward. A TAMER agent learns a "good" policy faster than a traditional reinforcement agent learner, but the latter is better at maximizing the final, peak performance after many more trials. To climb up the learning curve, in the TAMER+RL framework [22,24], the agent learns from both the human and environmental feedback, leading to a better performance than learning from either alone. This can be done sequentially (i.e., the agent first learns from the human feedback and then the environmental feedback) [22] 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 [24]. TAMER+RL differs from shaping with potential-based rewards [33] in that it uses the output of the learned human reward function not the shaping reward from a pre-defined potential function.

Similar to the TAMER framework, Pilarski et al. [34] proposed a continuous action actor-critic reinforcement learning algorithm that learns an optimal control policy for a simulated upper-arm robotic prosthesis using only human-delivered reward signals. Their algorithm does not model the human reward signals but treats them the same as the environmental rewards in traditional RL and tries to learn a policy to receive the most discounted accumulated human reward similar to non-myopic TAMER [25]. In their experiment, a human user teaches the robot arm to learn a control policy that outputs joint velocity commands to match three activities—reaching, retracted and relaxed, with each consisting of a temporal sequence of

two target joint angles—wrist and elbow. The state space consists of the two joint angles and two differentials of measures of electrical activity for each pair of opposing arm muscle groups, and the actions consists of two angular velocities.

While the work mentioned above interprets human feedback as a numeric reward, Loftin et al. [30] interpret human feedback as categorical feedback strategies that depend both on the behavior the trainer is trying to teach and the trainer's teaching strategy. They infer knowledge about the desired behavior from cases where no feedback is provided and show that their algorithms could learn faster than algorithms that treat the feedback as a numeric reward.

In addition, Griffith et al. [12] proposed an approach called 'policy shaping' by formalizing the meaning of human feedback as a label on the optimality of actions and using it directly as policy advice, instead of converting feedback signals into evaluative rewards. They estimate the probability of optimality of a state-action pair and the feedback policy from human feedback with a proposed Bayes optimal algorithm—Advise. The estimated feedback policy is combined with the learned policy from traditional reinforcement learning (in this case, it is Bayesian Q-learning [7] which learns a distribution for each Q value and uses it to estimate the probability that each action a is optimal in a state s). They compare the learning performance with policy shaping to strategies of combining human feedback and environment rewards developed in TAMER+RL and traditional reward shaping [33] in a series of experiments using a simulated human teacher. Their results show that Advise has similar performance to these state of the art methods, but is more robust to infrequent and inconsistent feedback. Cederbog et al. [6] extended their work by evaluating policy shaping with real human teachers and show that policy shaping is suitable for human generated data and participants in the experiment even outperform the simulated teacher because they are able to recognize multiple winning policies. In addition, they evaluate the effect of verbal instructions for when to give feedback or be silent and different interpretations of silence. Their results show that the quality of data is affected by what instructions are given to the human teacher and how the data is interpreted.

In general, an agent learning from human reward could learn a policy that outperforms the policy that the trainer intends to teach on the performance metric of the task that the trainer is attempting to maximize. However, the agent learning is still limited by the quality of the interaction between the human trainer and agent. Most above work focused on how to facilitate an agent to learn from ordinary people with human reward or using human reward to speed up an agent's learning in a pre-defined task, but does not investigate how to improve the agent's learning from human reward as we do in this article. However, Knox and Stone do investigate methods to improve the agent's learning from human reward using environmental rewards via a pre-defined reward function in the TAMER+RL framework [22,24]. Our work in this article differs in that it comes from the perspective of the interaction between the agent and human trainer.

Our preliminary work on this topic was presented in [27,28]. This article significantly extends upon our initial work by providing a more extensive analysis of the effect of agent's social competitive feedback on the human trainer's training behavior and agent learning performance and the real effect of the additional active socio-competitive feedback. Our results of the analysis of agent's learning performance with both the original score mechanism (number of lines cleared) and traditional Tetris game score mechanism (with bonuses) show that only half of trainers in the active social condition were affected by the agent's active social feedback, and agents trained by them finally outperformed those trained by participants affected by only passive social feedback. Moreover, our analysis shows that under the traditional Tetris game score mechanism (with bonuses), agents trained by participants affected by both passive and active social competitive feedback learned policies that can obtain a higher score than just counting the number of lines cleared and this effect started

from the beginning of the training and increased along the training process. However, the effect of agent's performance under the traditional Tetris game score mechanism plateaued on participants affected by the agent's passive social feedback, while the agent's additional active social feedback could further keep some trainers training agents to reach a higher performance.

2.2 Social networks

Research on Online Social Networks (OSNs) emanates from a wide variety of disciplines and involves research such as the descriptive analysis of users, motivations for using Facebook, identity presentation, the role of Facebook in social interactions, and privacy and information disclosure [44].

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. [32] created three popular applications with Facebook Developer, a platform for developers to build social apps on Facebook. These apps had over 8 million users, providing an enormous data set for research. In addition, in a social game also on Facebook, Kirman et al. [19] allow the game to display social-contextual information (e.g. centrality, density, reciprocity, etc.) extracted from the pattern of communication among users within the game. They find that these additional socio-contextual information can increase the social interactive activity and engagement of the users. Similar to this article, Rafelsberger and Scharl [35] propose an application framework to develop interactive games with a purpose on top of social networking platforms, leveraging the wisdom of the crowd to complete tasks that are trivial for humans but difficult for computers. In their game, they used the functionality of the platform's API to motivate the users to play the game and recruit new users, e.g., incentives such as awards, prizes, and high score ranking, users inviting friends to install the application will receive a certain percentage of the points, and leveraging the viral notification system to spread information of the application among the users of the network. In this article, we leverage the social aspects of OSNs both to recruit human trainers and to increase the users' engagement, thus improving the performance of the agents they train.

2.3 Gamification

Gamification is defined as the use of game-design elements (such as a score or leaderboard) in non-game contexts [8]. 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 [18,45].

Hamari et al. [14] survey 24 empirical studies on gamification. Their review indicates that, while most papers report positive effects of gamification, those effects are greatly dependent on the context in which the gamification is implemented, as well as on the users using it. The gamification implementations varied in terms of what game-like motivational affordances were implemented. Ten different motivational affordances, like points, leaderboards, badges, levels, theme, clear goals etc., were tested in the surveyed papers, with points, leaderboards, and badges being the most commonly used. Therefore, in our work, we choose points (scores) and a leaderboard—the two most commonly used motivational affordances—to implement in our system.

Hamari et al. found that a wide range of gamification contexts have been considered, such as commerce, education/learning, health/exercise, intra-organizational systems etc. For example, studies in the context of intra-organizational systems investigated the effects of gamifying IBM's Beehive system at different stages [11,40]. The main results from these studies indicate that gamification has a positive effect on some users for a short time [11]. Among all contexts using gamification, education or learning was the most common context for the implementations, e.g., using game-like elements to help teach users how to code or use software [9,10,13]. All studies in education/learning contexts found mostly positive learning outcomes from using gamification to increase motivation, engagement and enjoyment in the learning tasks. At the same time, the studies also pointed to some negative outcomes, such as the effects of increased competition and task evaluation difficulties. For instance, Dominguez et al. [9] 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 processor, spreadsheet), but performed poorly on written assignments and participated less in class activities. The context of education/learning with gamification is most similar to the work in this article. However, our work differs in that, by gamifying the teaching process, we are trying to motivate the teachers to see how gamification affects the teachers' behavior and the resultant learner's performance. In contrast, previous work tried to motivate the learners to see how gamification affects the learners' behavior and performance. Moreover, in our case, the learners are agents, while in previous work the learners are human beings.

Inspired by previous work on gamification, in this article, 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.

3 Background

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

3.1 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 [25], general myopia was a feature of all algorithms involving learning from human feedback and has received empirical support. Built to solve a variant of a Markov decision processes,

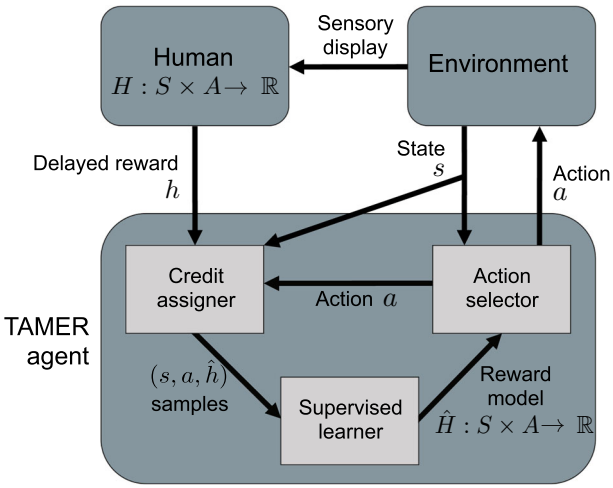

Fig. 1 Interaction in the TAMER framework (reproduced from [20])

(i.e., a specification of a sequential decision-making problem commonly addressed through reinforcement learning [37]) in which there is no reward function encoded before learning, the TAMER agent learns a function $\hat{R}_H(s, a)$ that approximates the expectation of experienced human reward, $R_H : S \times A \rightarrow \Re$. Given a state s, the agent myopically chooses the action with the largest estimated expected reward, $\arg\max_a \hat{R}_H(s, a)$. The trainer observes the agent's behavior and can give reward corresponding to its quality.

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

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 and the label for a sample is calculated as a delay-weighted aggregate reward based on the probability that a human reward signal targets a specific time step [20]. The TAMER learning algorithm repeatedly takes an action, senses reward, and updates $\hat{R}_H$. Note that unlike [20], when no feedback is received from the trainer, learning is suspended until the next feedback instance is received.

3.2 Experimental platform: tetris

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]

Tetris is played on a $10(w) \times 20(h)$ game board, in which seven different shapes of Tetris piece, called *tetrominoes*, composed of multiple configurations of four blocks are selected

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

randomly and fall from the top of the game board. A player can configure the horizontal placement of consecutive blocks at the base of the board or on top of previously placed pieces. When a row is completely filled with blocks, all the blocks in that row are cleared, and the remaining blocks above this row fall to fill the cleared line. The game ends when the blocks stack beyond the top of the grid. During this task, the player's goal is to arrange the pieces so as to clear as many lines as possible before the game ends.

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 2^{200}. In the TAMER framework, the agent uses 46 state features—including the ten column heights, nine 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 [20]—to represent the state observation. The input to $\hat{R}_H$ is 46 corresponding state-action features, the difference between state features before a placement and after the placement and clearing any resulting solid rows.

Like other implementations of Tetris learning agents (e.g., [3,5,38]), the TAMER agent only chooses an action from a set of possible final placements of a piece on the stack of previously placed pieces. That is, for a given action, the combination of atomic rotations and left/right movements of a piece to place it in the chosen position are determined independently by the agent and not learned via trainer feedback. Even with this simplification, playing Tetris remains a complex and highly stochastic task.

4 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 article, 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 (allowing to use the user information such as email address, location, age, gender, etc.), terms and conditions (policy regarding the privacy and personal data protection) to authenticate the app. As shown in Fig. 2, the training page contains a game board on the left side and a tip box on the right that shows training instructions.

The key advantage of the Socio-competitive TAMER interface is that, as with other experimental uses of Facebook [19,32,35], 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 [29] 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.

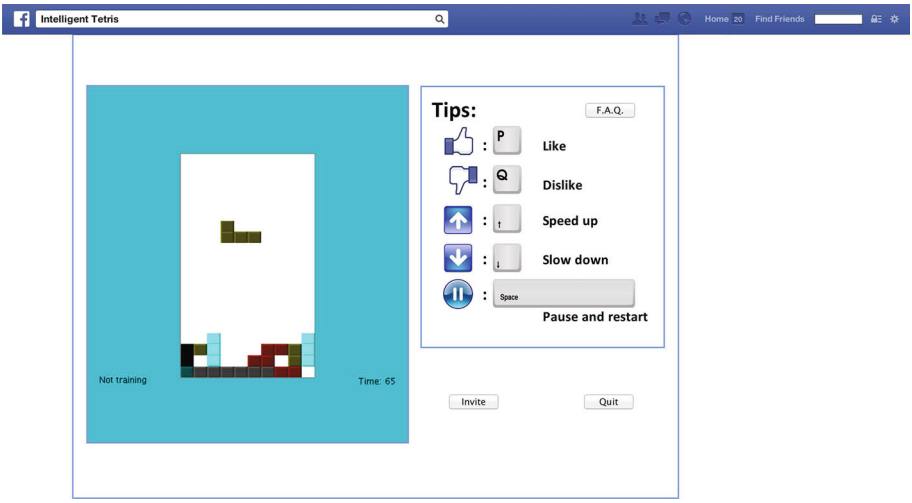

Fig. 2 Intelligent Tetris, our Facebook app for training. This is the interface used in the control condition described in Sect. 5, in which the trainer is not given any feedback except for the state and action of the agent

5 Experiment conditions

In this section, we present the four conditions used in our experiment. The control and performance conditions are replicated from our earlier work [26,29]. The passive and active social conditions are the novel conditions we propose in this article. The conditions and their corresponding functionalities are summarized in Table 1. As described below, in Table 1, 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 participants.

5.1 Control condition

The interface for the *control* condition, shown in Fig. 2, is the original TAMER interface presented in [21] but placed within the Socio-competitive TAMER interface, as described in Sect. 4. 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.

Table 1 Summary of the four conditions

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

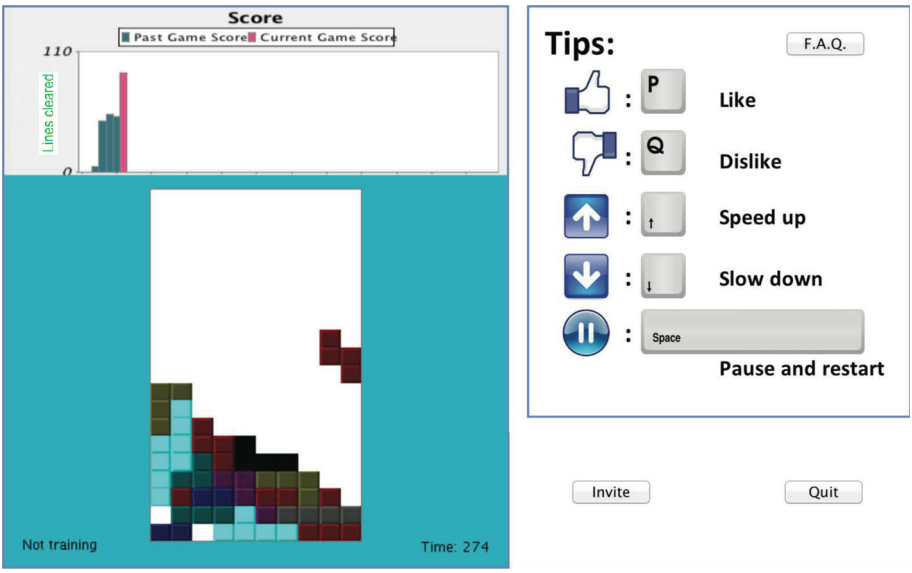

Fig. 3 The performance interface. Unlike in the control condition shown in Fig. 2, a performance window indicating the agent's performance per game in the learning process is added on top of the game board

5.2 Performance condition

The *performance* condition is implemented by integrating the performance-informative interface of [29] 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 Fig. 3, 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 [26,29] 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.

5.3 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.

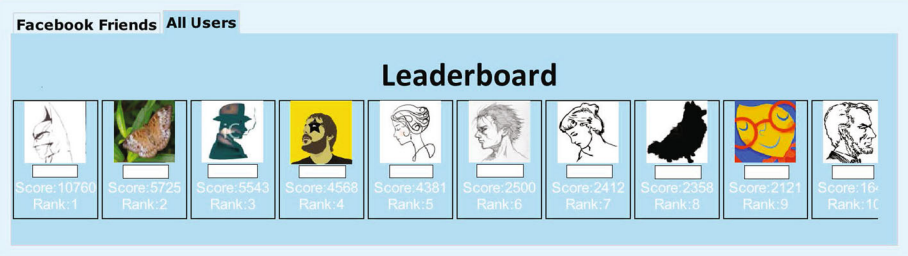

Fig. 4 The leaderboards. (The profile images and names are obscured for anonymization reasons)

To implement this condition, we added a leaderboard on top of the interface of the *performance* condition. An example leaderboard is shown in Fig. 4. 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 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 in 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 off-line 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.

5.4 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 Fig. 5. 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.

Fig. 5 Notification (with names anonymized)

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 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 to avoid sending too many notifications and keep relatedness, 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 trainers (i.e., ranks in the 'All Users' leaderboard) 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 Fig. 5.

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 (i.e., only the first three notifications within one day will be sent).

6 Experimental setup

To evaluate our interfaces, we conducted an experiment with our 'Intelligent Tetris' Facebook App. For recruiting participants, we advertised our experiment and Facebook App in the university via putting up posters and posting messages on our Facebook pages to share with friends and colleagues. People can also invite their friends via an 'Invite' button on the interface once they joined in. One hundred and fifty-seven 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]

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

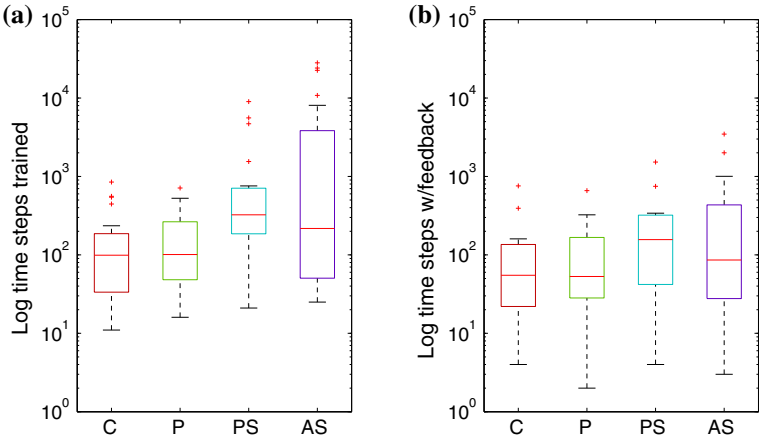

Fig. 6 *Boxplots* across the four conditions of **a** total time steps trained by participants and **b** between-subject distribution of the total number of time steps that were labeled with feedback. *C* Control, *P* performance, *PS* passive social, *AS* active social

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 [26,29], the trainers were not given time to practice before training started.

7 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 indicated with '+') if they are 1.5× IQR larger than the third quartile or 1.5× IQR smaller than the first quartile.

7.1 Training time

Figure 6a 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. Table 2 summarizes the mean and median of the total number of time steps over all participants in each condition and the p values for comparing the passive social and active social condition with the control and performance conditions. Since the leaderboard is the primary social feature in both passive social and

Table 2 The mean and median of the total number of time steps trained by participants in each condition, as well as the p values from comparing the passive social, active social and passive+active social conditions with control and performance conditions

	Mean	Median	p value	
			Control	Performance
Control	185	99		
Performance	177	101		
Passive social	1260	324	**<0.01**	**<0.02**
Active social	4300	217	0.05	0.08
Passive+active social	2949	244	**<0.01**	**<0.03**

Note that passive+active social condition is the set of all participants in passive and active social conditions. Bold values indicate statistical significance ($p < 0.05$)

active social conditions, we also show the mean and median total number of time steps in the passive+active social condition by putting all participants in both social conditions together in the table.

The results show that, in the passive social conditions, the participants 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; $U = 110.5$, $z = -2.22$, $p < 0.02$, $r = 0.36$). In the active social condition, the participants also had a trend to train 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; $U = 179$, $z = -1.37$, $p = 0.08$, $r = 0.21$), though lacking statistical significance. From these results we can see that in the passive social condition, the agent's passive social feedback can influence trainers to train statistically significant longer than both the control and performance conditions, while in the active social condition, the agent indicating both passive and active social feedback had trainers a trend to train longer than both control and performance conditions.

Finally, Table 2 shows that, in the passive+active social condition, participants trained significantly longer than in either the control (2.46 times in median; $U = 302.5$, $z = -2.33$, $p < 0.01$, $r = 0.29$) or performance conditions (2.41 times in median; $U = 289.5$, $z = -2.02$, $p < 0.03$, $r = 0.25$). Thus, the social conditions positively affected training time, which is consistent with our hypotheses overall.

7.2 Amount of feedback

Figure 6b summarizes the distribution of the number of time steps with feedback for all the participants in the four conditions. Table 3 summarizes the mean and median number of time steps with feedback over all participants in each condition, as well as the p values for comparing the social conditions with the control and performance conditions. The results show that, in the passive social condition, the trainers had a trend to give more feedback than in the control (2.9 times in median; $U = 152.5$, $z = -1.49$, $p = 0.07$, $r = 0.23$), though not statistically significant.

In addition, Table 3 also shows that in the passive+active social condition, participants had a trend to give more feedback than in the control condition (1.6 times in median; $U = 354.5$, $z = -1.62$, $p = 0.05$, $r = 0.20$), though lacking statistical significance. Overall, though our results show that the social conditions positively affected the number of time steps with feedback, the results are not statistically significant. We believe that the lack of statistical

Table 3 The mean and median number of time steps with feedback trained by participants in each condition, as well as the p value for comparing the passive social, active social and passive+active social conditions with the control and performance conditions

	Mean	Median	p value	
			Control	Performance
Control	112.76	55		
Performance	122.10	53		
Passive social	253.95	156.5	0.07	0.098
Active social	416	86	0.09	0.18
Passive+active social	343.98	86	0.05	0.11

Note that passive+active social condition is the set of all participants in passive and active social conditions.

Table 4 The mean and median total instances of feedback given by trainers for each condition, as well as the p value for comparing the passive social, active social and passive+active social conditions with control and performance condition

	Mean	Median	p value	
			Control	Performance
Control	243.47	92		
Performance	372.95	131		
Passive social	615.7	293	**<0.03**	0.11
Active social	665.08	181	0.07	0.18
Passive+active social	643.13	225	**<0.03**	0.11

Note that passive+active social condition is the set of all participants in passive and active social conditions. Bold values indicate statistical significance ($p < 0.05$)

significance could be because only some participants were affected or that the sample size is too small.

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. Table 4 summarizes the mean and median total instances of feedback given by all participants in each condition, as well as the p values for comparing the social conditions with the control and performance conditions. In the passive social condition, the trainers gave significantly more feedback instances (number of times positive or negative feedback button was pressed) than in the control condition (3.2 times in median; $U = 136.5$, $z = -1.90$, $p < 0.03$, $r = 0.30$). In the active social condition, the trainers had a trend to give more feedback instances than in the control condition (2.0 times in median; $U = 194.5$, $z = -1.49$, $p = 0.07$, $r = 0.22$), though lacking statistical significance. However, Table 4 shows that in the passive+active social condition, participants gave significantly more feedback instances than in the control condition (2.45 times in median; $U = 331$, $z = -1.94$, $p < 0.03$, $r = 0.24$).

In summary, our results suggest that, overall, the agents' social performance feedback can influence the trainer to spend more time on training.

Table 5 The mean and median final offline performance measured by counting the number of lines cleared, for each condition, as well as the p value for comparing the passive social, active social and passive+active social conditions with control and performance condition

	Mean	Median	p value	
			Control	Performance
Control	270.8	3.95		
Performance	272.1	5.15		
Passive social	668.6	18.1	0.20	0.48
Active social	590.8	19.93	**0.03**	0.12
Passive+active social	625.4	19.93	0.05	0.22

Note that passive+active social condition is the set of all participants in passive and active social conditions. Bold value indicates statistical significance ($p < 0.05$)

7.3 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 trainer, 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 levels across conditions, the domain stochasticity etc., may still affect the evaluation of agent performance. Nonetheless, we believe that the relatively 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 the trainers in each condition. If the participant'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 [26,29], in the analysis below we use the mean value across trainers as the performance for each condition. Table 5 summarizes the mean and median final offline performance for the four conditions and the passive+active social condition—a combination of passive and active social conditions, and highlights the positive effect of the agent's social feedback on its learning performance.

As shown in Fig. 7, early performance within the four conditions was similar to each other, with the performance in the active social condition a bit better. Thereafter, the performance within the passive social and active social conditions increase faster than the control and performance conditions. 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 ($p = 0.23$). There are two possible explanations that may answer it: one is that the effect of the active social feedback could be minor in our experiment and the other explanation can be that the trainers might train a more complex agent behavior by optimizing on their own performance metric.

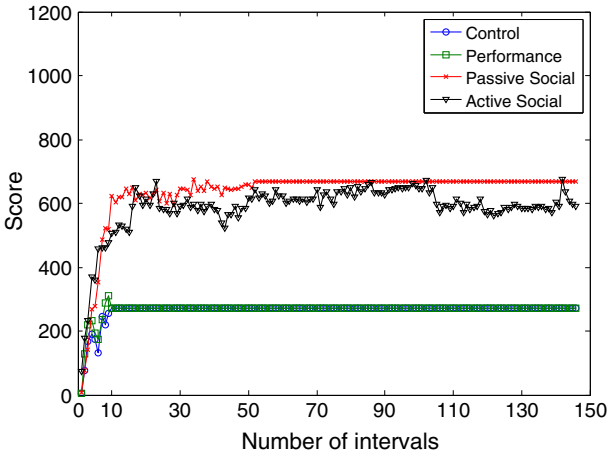

Fig. 7 Mean offline performance for the four conditions during training. Note that the score is measured by counting the number of lines cleared

7.3.1 Minor effect of active social feedback

One explanation of why agents in the active social condition did not outperform those in the passive social condition can be that the improvement of the active social condition is relatively small and cannot be detected in our experiment, given the size of the groups and variation in the trainers, or only a small proportion of trainers in the active social condition were affected. From the results in Sect. 7.1 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. This can be clearly seen from Fig. 6 which shows in the active social condition trainers trained agents with considerably more time steps on average than in the passive social condition (Fig. 6a), but this reduces in the time steps with feedback (Fig. 6b). Therefore, the reason why agents in the active social condition did not outperform those in the passive social condition can be that the effect of active social feedback on the trainers' training and agent's learning is minor and a small proportion of trainers in the active social condition were affected. We will further examine this later in Sect. 7.4.2.

7.3.2 Multi-line clearing strategy

Another possible explanation we considered is 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 clearing 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 for the traditional Tetris game

gives increasing bonuses (0.5, 4.5, 26) for clearing 2, 3, or 4 lines respectively. We found that the new final offline performance for the *active social* condition increased more (from 590 to 716 lines) than the new final offline performance of the *passive social* condition (from 668 to 742 lines), when compared to their corresponding bonus-free performances.

This difference in training behavior between the two conditions was not expected and we do not know why the trainers in the *active social* condition stopped training before their agent performance surpassed those of the *passive social* condition. A possible explanation is that the *active social* trainers stopped training because their rankings were not improving even with their more complex training strategy. 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.

7.4 Influence of social information

7.4.1 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). This suggests that the agent's social competitive feedback can motivate trainers to train the agents longer and better, which supports our results in Sects. 7.1 and 7.3.

7.4.2 Receiving notifications

The data shows that 22 trainers received 40 notifications in total in the *active social* condition. Of them, 13 trainers received one notification. Of the 40 notifications, 9 notifications said she had been surpassed by others and 31 notifications informed the trainer she surpassed other trainers. Of the 22 trainers receiving notifications, 2 trainers only received notifications saying she had been surpassed by others, 16 of them only received notifications informing the trainer she surpassed other trainers, and 4 of them received both. The notification jewel was clicked 28 times in total by 11 of the 22 trainers (each trainer at least once), 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 ($r = 0.18$, $p = 0.39$) though not statistically significant and the number of time steps with feedback ($r = 0.41$, $p = 0.04$), and the number of clicks correlates with the number of time steps with feedback ($r = 0.43$, $p = 0.03$) and the trained agents' final offline performances ($r = 0.44$, $p = 0.03$). These results indicate that the agent's active social feedback can motivate trainers to train agents with more feedback and better, which also supports our results in Sects. 7.1 and 7.3.

However, although most trainers in the active social condition received notifications, 13 of them received only one notification and only 11 trainers clicked on the notification and came back for further training. Therefore, only 11 of the 25 trainers in the active social condition were affected by the agent's active social feedback, and the remaining 14 trainers behaved in the same way as those in the passive social condition. We will provide additional analysis about the real effect of the agent's active social feedback by moving the 14 trainers from the active social condition in the subsequent section. Moreover, we also suspect that the effect of active social feedback is related to the number and kind of notifications received. Intuitively, a notification saying a participant is beaten by others will motivate the participant to come back and further train the agent. In the active social condition, only 6 trainers received notifications saying she had been surpassed by others and 4 of them who received both kinds of notifications trained agents performing much better than most trainers in the active social condition. Furthermore, the 4 trainers who received both kinds of notifications also clicked on the notification for at least once. This could explain why 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), and gave less feedback in median (0.6 times) but more mean time steps with feedback than in the passive social condition (1.6 times), as discussed in Sects. 7.1 and 7.2. In addition, the results support the reason why agents in the active social condition did not significantly outperform those in the passive social condition, since the effect of the agent's active social feedback is relatively small and only a small proportion of participants in the active social conditions were affected.

7.5 Real effect of active social feedback

As we found in Sect. 7.4, only 11 participants in the active social condition were influenced by the agent's active social feedback to click on the received notifications and come back for further training. Therefore, we keep the 11 trainers who clicked on notifications in the active social condition and call it the *actual active social condition*. Then we move the remaining 14 trainers to the passive social condition since 3 of them did not receive notifications at all and the other 11 received notifications but were not influenced by the agent's notifications to come back for further training, i.e., they behaved in the same way as those participants in the original passive social condition. We then rename this condition the *modified passive social condition* (passive social + passive in active social condition). Figure 8a summarizes the total number of time steps trained for the modified passive social condition and actual active social condition (note the log scale). Figure 8b summarizes the distribution of the number of time steps with feedback for all the participants in the the two conditions, which is similar to the distribution in the original passive social and active social conditions in Fig. 6.

Then, to reduce the effect of the stochasticity of the Tetris domain, we retested each recorded policy for 200 games with both the original score mechanism (i.e., counting the number of lines cleared, which is the same as the mechanism used in Fig. 7) and the traditional Tetris game score (with the same bonuses for lines cleared in Sect. 7.3). The policy was recorded at intervals divided up in the same way as in Sect. 7.3. The offline performance for each policy is averaged over the 200 games. The performance for each condition is also calculated in the same way as in Sect. 7.3.

We report the offline performance along the whole training process in Fig. 9. As shown in Fig. 9, we found that with both score mechanisms, agents in the *actual active social condition* performed similar to those in the *modified passive social condition* in the early training stage. However, agents in the *actual active social condition* outperformed those in the *modified passive social condition* from about 44 intervals on until the end of training ($p < 0.08$ for all

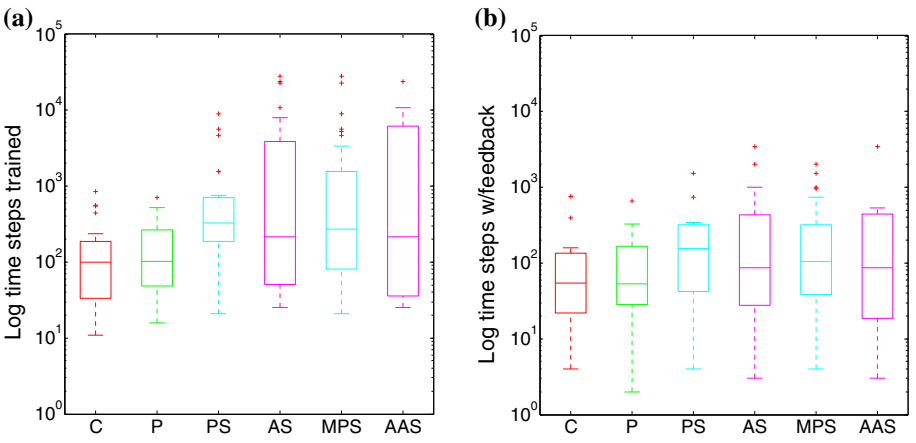

Fig. 8 *Boxplots* across the modified passive social (MPS) and actual active social (AAS) conditions of **a** total time steps trained by participants and **b** between-subject distribution of the total number of time steps that were labeled with feedback, in comparison with the original four conditions, *C* control, *P* performance, *PS* passive social, *AS* active social

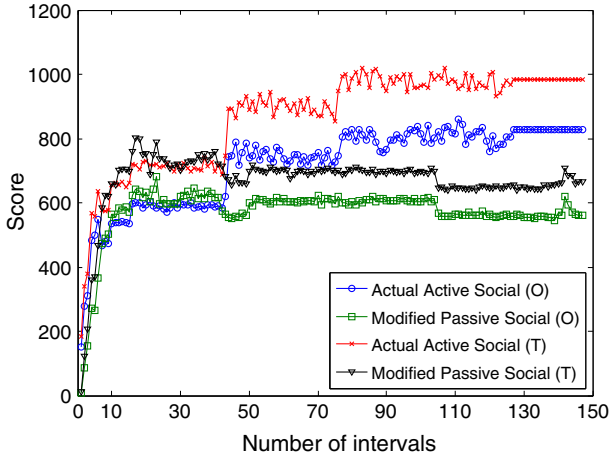

Fig. 9 Mean offline performance with the original score mechanism (O) and traditional Tetris game score mechanism (T) for the Actual Active Social Condition and the Modified Passive Social Condition (after moving participants not affected by the agent's active social feedback to the original passive social condition) during training. Note that for the original score mechanism (O) which is also used in Fig. 7, the score is measured by counting number of lines cleared; for the traditional Tetris game score mechanism (T), the score is measured with increasing bonuses (0.5, 4.5, 26) for learning *2, 3, 4 lines* respectively, as in the traditional Tetris game

intervals thereafter), which may indicate a change in line clearing strategy. Moreover, Fig. 9 shows that the performance in the modified passive social condition plateaued after about 15 intervals and kept at a similar level until the end of training. These results show that some trainers in the actual active social condition were affected by the agent's active feedback and came back to successfully train agents to perform better.

To see whether the performance difference is caused by successfully training a multi-line clearing strategy and when this strategy was trained, we plotted the score difference between

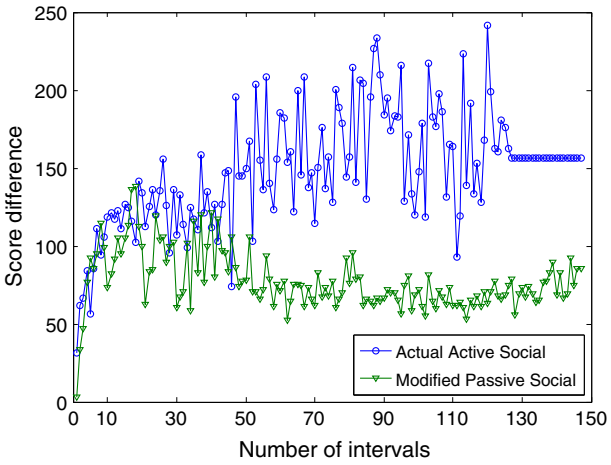

Fig. 10 The score difference of mean offline performance between scores measured with the original score mechanism (counting number of *lines* cleared) and traditional Tetris game score mechanism (with bonuses) for the Actual Active Social Condition and Modified Passive Social Condition (after moving participants not affected by the agent's active social feedback to the original passive social condition) during training

the two score mechanisms for the *actual active social condition* and *modified passive social condition* along the whole training process in Fig. 10. As we found in Fig. 10, agents trained by participants in both the *actual active social condition* and *modified passive social condition* obtained a higher score with the original Tetris game score mechanism than just counting the number of lines cleared. The score difference between these two score mechanisms increases from the beginning of training and are similar in both conditions at the early training stage. This result indicates that the effect of multi-line clearing strategy (under the score mechanism with bonus) started from the beginning and increased along the training process. However, the effect of the multi-line clearing strategy in the *modified passive social condition* plateaued in about 9 intervals and kept at a similar level until the end of training, while the effect in the actual active social condition was still slightly increasing and kept higher than that in the *modified passive social condition* until the end of training. This result suggests that the agent's additional active social feedback may encourage a time varying training strategy and motivate participants to further keep training a policy that can obtain a higher performance under a multi-line clearing strategy.

8 Conclusion

By integrating agent training with an online social network via the Socio-competitive TAMER interface, this article investigates 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 show 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 is much better when socio-competitive feedback is provided to the trainer.

In addition, we find that adding active social feedback induces more trainers to train longer (on average, but less in median) and provide more feedback and further improves the agent's

learning. Our further analysis also suggests that under a multi-line clearing strategy that could be optimized from the trainer's perspective, agents trained by participants affected by both the agent's passive and active social feedback can obtain a higher game score than just counting lines cleared, and this effect increases from the beginning of training. However, the effect of the multi-line clearing strategy plateaued for participants affected by the agent's passive social feedback, while the agent's active social feedback could further keep participants training a policy that obtained a higher performance under a multi-line clearing strategy. The results in this article suggest that while the interaction design for agent training is important, a performance metric for the task defined from the human teacher's perspective will be useful for the training process. 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. [20].

Acknowledgements We would like to thank the anonymous reviewers and editors for their helpful comments. This work was partially supported by the Fundamental Research Funds for the Central Universities (under Grant No. 841713015) and China Postdoctoral Science Foundation.

References

1. Argall, B., Browning, B., & Veloso, M. (2007). Learning by demonstration with critique from a human teacher. In *Proceedings of the ACM/IEEE international conference on human–robot interaction* (pp. 57–64). ACM.
2. Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, *57*(5), 469–483.
3. Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming* (Vol. 7). Optimization and neural computation series 3. Belmont, MA: Athena Scientific.
4. Blumberg, B., Downie, M., Ivanov, Y., Berlin, M., Johnson, M. P., & Tomlinson, B. (2002). Integrated learning for interactive synthetic characters. In *ACM transactions on graphics (TOG)* (pp. 417–426). ACM.
5. Böhm, N., Kókai, G., & Mandl, S. (2004). Evolving a heuristic function for the game of tetris. In *LWA* (pp. 118–122).
6. Cederborg, T., Grover, I., Isbell, C. L., & Thomaz, A. L. (2015). Policy shaping with human teachers. In *Proceedings of the 24th international conference on artificial intelligence* (pp. 3366–3372). AAAI Press.
7. Dearden, R., Friedman, N., & Russell, S. (1998). Bayesian q-learning. In *AAAI/IAAI* (pp. 761–768).
8. Deterding, S., Khaled, R., Nacke, L. E., & Dixon, D. (2011). Gamification: Toward a definition. In *CHI 2011 gamification workshop proceedings* (pp. 12–15).
9. Domínguez, A., Saenz-de Navarrete, J., De-Marcos, L., Fernández-Sanz, L., Pagés, C., & Martínez-Herráiz, J.-J. (2013). Gamifying learning experiences: Practical implications and outcomes. *Computers & Education*, *63*, 380–392.
10. Dong, T., Dontcheva, M., Joseph, D., Karahalios, K., Newman, M., & Ackerman, M. (2012). Discovery-based games for learning software. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 2083–2086). ACM.
11. Farzan, R., DiMicco, J. M., Millen, D. R., Dugan, C., Geyer, W., & Brownholtz, E. A. (2008). Results from deploying a participation incentive mechanism within the enterprise. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 563–572). ACM.
12. Griffith, S., Subramanian, K., Scholz, J., Isbell, C., & Thomaz, A. L. (2013). Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in neural information processing systems* (pp. 2625–2633).

13. Hakulinen, L., Auvinen, T., & Korhonen, A. (2013). Empirical study on the effect of achievement badges in TRAKLA2 online learning environment. In *Learning and teaching in computing and engineering (LaTiCE), 2013* (pp. 47–54). IEEE.
14. Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does gamification work?—A literature review of empirical studies on gamification. In *2014 47th Hawaii international conference on system sciences (HICSS)* (pp. 3025–3034). IEEE.
15. Isbell, C., Shelton, C. R., Kearns, M., Singh, S., & Stone, P. (2001). A social reinforcement learning agent. In *Proceedings of the fifth international conference on autonomous agents* (pp. 377–384). ACM.
16. Isbell, C. L, Jr., Kearns, M., Singh, S., Shelton, C. R., Stone, P., & Kormann, D. (2006). Cobot in LambdaMOO: An adaptive social statistics agent. *Autonomous Agents and Multi-agent Systems*, *13*(3), 327–354.
17. Kaplan, F., Oudeyer, P.-Y., Kubinyi, E., & Miklósi, A. (2002). Robotic clicker training. *Robotics and Autonomous Systems*, *38*(3), 197–206.
18. Kapp, K. M. (2012). *The gamification of learning and instruction: Game-based methods and strategies for training and education*. Hoboken: Wiley.
19. Kirman, B., Lawson, S., Linehan, C., Martino, F., Gamberini, L., & Gaggioli, A. (2010). Improving social game engagement on Facebook through enhanced socio-contextual information. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1753–1756). ACM.
20. Knox, W. B. (2012). Learning from human-generated reward. Ph.D. thesis, University of Texas at Austin.
21. Knox, W. B., & Stone, P. (2009). Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the fifth international conference on knowledge capture* (pp. 9–16). ACM.
22. Knox, W. B., & Stone, P. (2010). Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In *Proceedings of the 9th international conference on autonomous agents and multiagent systems* (pp. 5–12). International Foundation for Autonomous Agents and Multiagent Systems.
23. Knox, W. B. & Stone, P. (2012). Reinforcement learning from human reward: Discounting in episodic tasks. In *2012 IEEE on RO-MAN* (pp. 878–885). IEEE.
24. Knox, W. B. & Stone, P. (2012). Reinforcement learning from simultaneous human and MDP reward. In *Proceedings of the 11th international conference on autonomous agents and multiagent systems* (pp. 475–482). International Foundation for Autonomous Agents and Multiagent Systems.
25. Knox, W. B., & Stone, P. (2015). Framing reinforcement learning from human reward: Reward positivity, temporal discounting, episodicity, and performance. *Artificial Intelligence*, *225*, 24–50.
26. Li, G., Hung, H., Whiteson, S., & Knox, W. B. (2013). Using informative behavior to increase engagement in the TAMER framework. In *Proceedings of the 2013 international conference on autonomous agents and multi-agent systems* (pp. 909–916). International Foundation for Autonomous Agents and Multiagent Systems.
27. Li, G., Hung, H., Whiteson, S., & Knox, W. B. (2014). Learning from human reward benefits from socio-competitive feedback. In *Proceedings of the fourth joint IEEE international conference on development and learning and on epigenetic robotics* (pp. 93–100).
28. Li, G., Hung, H., Whiteson, S., & Knox, W. B. (2014). Leveraging social networks to motivate humans to train agents. In *Proceedings of the 2014 international conference on autonomous agents and multi-agent systems* (pp. 1571–1572). International Foundation for Autonomous Agents and Multiagent Systems.
29. Li, G., Whiteson, S., Knox, W. B., & Hung, H. (2015). Using informative behavior to increase engagement while learning from human reward. In *Autonomous agents and multi-agent systems* (pp. 1–23).
30. Loftin, R., Peng, B., MacGlashan, J., Littman, M. L., Taylor, M. E., Huang, J., & Roberts, D. L. (2015). Learning behaviors via human-delivered discrete feedback: Modeling implicit feedback strategies to speed up learning. In *Autonomous agents and multi-agent systems* (pp. 1–30).
31. Maclin, R., Shavlik, J., Torrey, L., Walker, T., & Wild, E. (1999, 2005). Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *Proceedings of the national conference on artificial intelligence* (p. 819). Menlo Park, CA: AAAI Press; Cambridge, MA: MIT Press.
32. Nazir, A., Raza, S., & Chuah, C.-N. (2008). Unveiling Facebook: A measurement study of social network based applications. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement* (pp. 43–56). ACM.
33. Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *ICML*, *99*, 278–287.
34. Pilarski, P. M., Dawson, M. R., Degris, T., Fahimi, F., Carey, J. P., & Sutton, R. S. (2011). Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *Proceedings of 12th international conference on rehabilitation robotics (ICORR)* (pp. 1–7). IEEE.
35. Rafelsberger, W., & Scharl, A. (2009). Games with a purpose for social networking platforms. In *Proceedings of the 20th ACM conference on hypertext and hypermedia* (pp. 193–198). ACM.

36. Suay, H. B., & Chernova, S. (2011). Effect of human guidance and state space size on interactive reinforcement learning. In *20th IEEE international symposium on robot and human interactive communication (RO-MAN)* (pp. 1–6). IEEE.
37. Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.
38. Szita, I., & Lörincz, A. (2006). Learning tetris using the noisy cross-entropy method. *Neural Computation*, *18*(12), 2936–2941.
39. Tenorio-Gonzalez, A. C., Morales, E. F., & Villaseñor-Pineda, L. (2010). Dynamic reward shaping: Training a robot by voice. In *Advances in artificial intelligence–IBERAMIA 2010* (pp. 483–492). Springer.
40. Thom, J., Millen, D., & DiMicco, J. (2012). Removing gamification from an enterprise SNS. In *Proceedings of the ACM 2012 conference on computer supported cooperative work* (pp. 1067–1070). ACM.
41. Thomaz, A. L., & Breazeal, C. (2006). Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. *AAAI*, *6*, 1000–1005.
42. Thomaz, A. L., & Breazeal, C. (2008). Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, *172*(6), 716–737.
43. Watkins, C. J., & Dayan, P. (1992). Q-Learning. *Machine Learning*, *8*(3–4), 279–292.
44. Wilson, R. E., Gosling, S. D., & Graham, L. T. (2012). A review of Facebook research in the social sciences. *Perspectives on Psychological Science*, *7*(3), 203–220.
45. Zichermann, G., & Linder, J. (2010). *Game-based marketing: Inspire customer loyalty through rewards, challenges, and contests*. Hoboken: Wiley.