A Need for Speed: Adapting Agent Action Speed to Improve Task Learning from Non-Expert Humans

Bei Peng
Washington State University
bpeng@eecs.wsu.edu

James MacGlashan
Brown University
jmacglashan@gmail.com

Robert Loftin
North Carolina State University
rtloftin@csc.ncsu.edu

Michael L. Littman
Brown University
mlittman@cs.brown.edu

David L. Roberts
North Carolina State University
robertsd@csc.ncsu.edu

Matthew E. Taylor
Washington State University
taylorm@eecs.wsu.edu

ABSTRACT

As robots become pervasive in human environments, it is important to enable users to effectively convey new skills without programming. Most existing work on Interactive Reinforcement Learning focuses on interpreting and incorporating non-expert human feedback to speed up learning; we aim to design a better representation of the learning agent that is able to elicit more natural and effective communication between the human trainer and the learner, while treating human feedback as discrete communication that depends probabilistically on the trainer's target policy. This work entails a user study where participants train a virtual agent to accomplish tasks by giving reward and/or punishment in a variety of simulated environments. We present results from 60 participants to show how a learner can ground natural language commands and adapt its action execution speed to learn more efficiently from human trainers. The agent's action execution speed can be successfully modulated to encourage more explicit feedback from a human trainer in areas of the state space where there is high uncertainty. Our results show that our novel adaptive speed agent dominates different fixed speed agents on several measures of performance. Additionally, we investigate the impact of instructions on user performance and user preference in training conditions.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Miscellaneous

General Terms

Languages, Performance, Human Factors, Experimentation

Keywords

Learning Sequential Decision Tasks from Humans; Human-Agent Interaction; Crowdsourcing Experiments; Eliciting Human Feedback; Variable Speed Agents

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

1. INTRODUCTION

As robots become pervasive in human environments, it is important to enable users to effectively convey new skills without programming. Recent work has shown that reinforcement learning [18] (RL) agents can successfully speed up learning using human feedback [3, 9, 20]. However, learning efficiently from human input brings significant challenges (e.g., delay, inconsistent input, and limited human patience) relative to learning autonomously.

This work is inspired by the ways humans train dogs — using rewards and punishments is a simple and familiar paradigm for humans to train intelligent agents. A human can easily correct the agent if it makes a mistake interpreting a command, rather than stopping the agent and providing a new language example. In particular, we treat human feedback as a discrete communication that depends probabilistically on the trainer's target policy. RL methods are most effective in environments where the numerically-valued reward function contains all the information needed to learn the policy. Learning from human feedback is substantively different. Trainers generally do not have numerical rewards to give to learners — instead, they use only a small set of discrete feedback signals and they may give those signals in a number of different ways to implicitly communicate the target behavior. Thus, while standard RL algorithms can be used in this setting, they are not designed to leverage all of the available information. Our previous work [12] showed that considering human feedback to be a discrete communication that depends probabilistically on the trainer's target policy can be more effective for training an agent, rather than treating it as a numerical reward signal as in traditional approaches. Our *Strategy-aware Bayesian Learning* (SABL) model is able to learn from both explicit and implicit feedback from human trainers.

While most existing work on Interactive Reinforcement Learning focuses on interpreting and incorporating human feedback to speed up learning, the design of agents is one neglected topic. We aim to design a better representation of the learning agent that is able to elicit more natural and effective communication between the human trainer and the learner, treating human feedback as a discrete value (reward, punishment, or neutral). Considering that human trainers get tired and then give infrequent feedback to the agent [11], trying to motivate people to provide more explicit feedback to the agent when it is in more important states is current work. To enable the agent to make the most of limited human input, higher amounts of explicit feedback should be delivered in areas of the state space where the learner has more uncertainty about how to act. Similarly, when the agent is fairly confident in its action choice, humans can expend less effort providing feedback.

To seek a more successful and effective learning interaction, the agent should aim to minimize the number of task failures, maximize its use of explicit human feedback, and minimize the total time needed to learn to successfully complete the task. The agent can move slower when it is unsure, giving human trainers more time to think about their strategy and deliver feedback. Similarly, when the agent is fairly confident in its action choice, it can move faster and finish the task more quickly. Lastly, the number of actions needed to learn a (near-)optimal policy should be minimized to reduce any per-action penalties (e.g., efficiently exploring a state space while minimizing wear on a robot). Note that minimizing the time and the number of actions is identical when the agent acts with a fixed time per action — if the action rate is variable, the number of actions executed can accrue at a different rate from the wall clock time expended.

Therefore, we are interested in designing an agent to optimize four metrics. We aim to 1) minimize the wall clock time the agent spends learning, 2) minimize the number of actions the agent takes while learning, 3) encourage the user to give explicit feedback to the agent (especially when the agent is unsure about its action choice), and 4) consistently learn a (near-)optimal policy.

In this paper, we present our Learning Agents Modeling Belief-Dependent Action Speeds (LAMBDAS) approach and results of a human-subjects experiment to show how a learner can adapt its action execution speed to learn more efficiently from human trainers. We demonstrate that the agent's action execution speed can be successfully modulated to strategically encourage more explicit feedback from a human trainer in parts of the state space where the learner has more uncertainty about how to act. Our results show that our novel adaptive speed agent dominates different fixed speed agents according to the four metrics discussed above.

2. BACKGROUND AND RELATED WORK

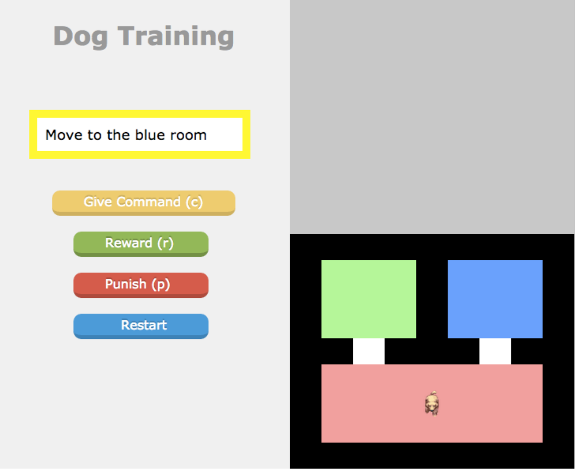
This section discusses the research most related to this paper's contributions.

There is a growing literature that addresses learning from human feedback [6, 7, 14]. Most existing reinforcement-learning algorithms treat human feedback as a numerical signal to be maximized by the agent. The TAMER algorithm is modeled after standard RL methods that learn a value function from human-delivered numerical rewards, and has been shown to be effective in a number of task domains [9, 10]. Thomaz and Breazeal [20, 21] treat human feedback as a form of guidance for an agent solving an RL problem. There, human feedback does not change the optimal policy for the RL problem, but improves exploration and accelerates learning. Their results show that humans give reward in anticipation of good actions, instead of rewarding or punishing the agent's recent actions. Suay and Chernova [17] extend their work by applying the Interactive Learning Algorithm to a real-world robotic system. They demonstrate that user guidance significantly reduces the learning rate and the impact of teacher guidance increases with state space size. Tenorio-Gonzalez et al. [19] introduce a dynamic reward shaping approach to learn from infrequent and inconsistent human feedback where the user's feedback is converted into a shaping reward, which is then used to modify the MDP reward to guide the agent's learning process.

Work by Griffith et al. [3] and Ho et al. [4] are perhaps the most similar to ours. Here, human feedback is treated as discrete communication that depends probabilistically on the trainer's target policy, rather than on some numerical reward, so that the trainer's feedback is making a direct statement about the policy itself rather than influencing the policy through a reward. Specifically, In the work of Ho et al. [4], instead of using the *reward-maximizing model* based on standard reinforcement learning, an *action-feedback model* is shown to be followed by the majority of people while teaching. However, the interaction between learning mechanism and feedback pattern is not explored. In the current paper, human feedback (either explicitly rewarding, explicitly punishing, or providing no feedback) after an agent's action is evidence for or against hypothesized target tasks represented by MDP reward functions. At each time step, the agent follows the optimal policy for the most likely task, which is computed using an "off-the-shelf" MDP planning algorithm.

To better learn from human feedback, we must deeply understand how do human teachers want to teach the agent. Work by Knox et al. [8] examine how users want to provide feedback, finding that: 1) there is little difference in a trainer's feedback whether they think that the agent can learn or that they are critiquing a fixed performance; and 2) humans can reduce the amount of feedback they give over time, and having the learner purposefully select bad actions can encourage more feedback. Similarly, our work addresses the problem of increasing the trainer's feedback quantity and quality. However, it is the first to consider modulating the agent's action execution speed according to its confidence level to elicit more explicit feedback. Our previous work [12] find that people choose different training strategies to teach the agent, and may use a lack of explicit feedback to communicate the same information as delivered by reward or punishment in both contextual bandit and sequential decision-making tasks. The choices of trainer strategies are shown to be affected by several factors (e.g., the nature of the training task or the learning agent). Instead of focusing on training strategies, we aim to seek a more successful and effective learning interaction, while maximizing the agent's learning performance on several measures. Our empirical user studies demonstrate that our novel adaptive speed agent can achieve this goal.

In this learning framework, the effect of different types of instructions on human feedback are also studied. Thomas et al. [2] show that human teachers are silent 30% of the time on average while providing feedback, implicating the importance for the policy shaping algorithm to interpret that silence. Different instructions are given to manipulate the meaning of silence to evaluate the performance impact. Pradyot et al. [15, 16] show that humans can provide sporadic instructions to speed up learning. The instructions here reduce the agent's search space and narrow down the potential interesting objects without specifying any information about how to perform the task. Similarly, we explore the impact of instructions on user performance. However,



Move to the blue room

Figure 1: The GUI used to train the agent with given commands. It shows the task used in the tutorial.

our instructions are not designed to manipulate trainer's feedback strategies or affect the agent's behavior. We want to understand whether users can interpret the behavior of the adaptive speed agent correctly and perform better if they have a correct belief about the agent's belief.

3. LANGUAGE LEARNING WITH REWARD AND PUNISHMENT

To enable people to naturally and effectively train an artificial agent to carry out a variety of different tasks with reward and punishment, one approach is for a person to manually specify each new task they are training and when they wanted an agent to perform a previously trained task, provide an interface that allowed them to select from the previously trained tasks. However, a more natural interface would be to connect the task learning with a natural language model. In this setting, a trainer could give a novel command and reward / punish the agent until the agent successfully completed the task. As the trainer taught additional tasks, the agent would become better at interpreting the language, thereby enabling the agent to successfully interpret and carry out novel commands without any reward and punishment. For example, an agent might learn the interpretation of "red" and "chair" from the command "move the red chair," and the interpretation of "blue" and "bag" from the command "bring me the blue bag," thereby allowing correct interpretation of the novel command "bring me the red bag." The domain we use in our experiment is a simplified simulated home environment. Our domain and user study GUI are shown in Figure 1.

To enable language learning from agents trained with reward and punishment, we use our previous probabilistic model [13] that connected the IBM Model 2 (IBM2) language model [1] with a factored generative model of tasks, and the goal-directed SABL algorithm [12] for learning from human feedback.

Using this probabilistic model, an iterative training regime proceeds as follows. First, the trainer gives an English command. Second, from this command, a distribution over the possible tasks for the current state of the environment is inferred using Bayesian inference. Third, this task distribution is used as a prior for the goals in goal-directed SABL. Fourth, the agent is trained with SABL for a series of time steps. Fifth, after completing training, a new posterior distribution over tasks is induced and used to update the IBM2 model via weakly-supervised learning. The process repeats when provided a new command.

To validate that the agent is able to successfully learn novel language commands using reward and punishment feedback from a human trainer, In our previous work [13] participants train the agent on seven commands including "take the blue chair to the yellow room" and "take the purple bag to the blue room." As training proceeds, the agent becomes better at correctly interpreting commands and requires less feedback to disambiguate the intended meaning. After training, the agent is able to correctly interpret novel commands that describe different combinations of objects, object colors, and room colors. For example, the agent correctly interprets the command "take the purple chair to the green room" despite never being trained with commands involving a purple chair or taking an object to a green room. However, in this work, we are concerned with the per-task training rather than validating the quality of the learned language model.

4. ADAPTIVE SPEED AGENT

This section presents our LAMBDAS approach to implement the adaptive speed agent in our experiment.

4.1 Motivation

Our goal is to design a framework in which non-expert humans (i.e., workers on Amazon's Mechanical Turk, also known as "Turkers") can provide categorical feedback to an agent to learn task groundings, while maximizing its learning performance. To seek a more successful and effective learning interaction, the agent should aim to minimize the number of task failures, maximize its use of explicit human feedback, and minimize the total time and number of actions needed to learn to successfully complete the task (discussed in Section 1).

Considering that human trainers tire and then give less feedback to an agent, it is a reasonable design decision to have the agent try to elicit explicit feedback when it is in some states. Higher amounts of explicit feedback should be delivered in parts of the state space where the learner has more uncertainty about how to act, so that the agent can make the most of limited human input. Similarly, when the agent is fairly confident in its action choice, humans can expend less effort providing feedback. Our core hypothesis was that changing the speed of the agent can be a good way to achieve this goal; the agent can move slower when it is unsure, giving the human trainers more time to think about the strategy and deliver feedback. Similarly, when the agent is fairly confident in its action choice, it can move faster and finish the task more quickly.

We are interested in studying whether users can interpret the behavior of the adaptive speed agent correctly and how user performance will change in response to the changing speed of the agent. Our insight was that the adaptive speed agent would be either competitive with or outperform the fixed speed agents according to the four metrics discussed above.

4.2 Modeling Belief-Dependent Action Speed

We develop the LAMBDAS approach to implement our novel adaptive speed agent. We first define the confidence level of the agent in its action choice, and then develop a model to map its confidence to different action execution speeds. The entropy (H) of the action selection is used to summarize the agent's confidence level,

$$H = -\sum_{a \in A} \Pr(a = a^*|s, F) \ln(\Pr(a = a^*|s, F)), \qquad (1)$$

where A is the set of possible actions, F is the history of feedback events from the trainer, and $\Pr(a = a^*|s, F)$ is the probability that action a given state s and feedback history F is the optimal action (a^*). A high entropy means the agent has high uncertainty in choosing the optimal action, while a low entropy indicates agent confidence in which action is the optimal action for the given state.

Because we follow the SABL formalism in which human feedback gives evidence for an intended reward function for which there are corresponding optimal policies, the probability that an action selection is the optimal action can be inferred by marginalizing the policies over the reward functions. Specifically, the probability distribution over optimal actions is computed as

$$\Pr(a = a^*|s, F) = \sum_{r \in R} \pi(s, a|r) \Pr(r|F), \qquad (2)$$

where R is the hypothesis space of possible reward functions to be trained, $\pi(s, a|r)$ is an indicator function specifying whether action a is the optimal action in state s when reward function r is being maximized,[1] and $\Pr(r|F)$ is the current probability based on the feedback history F that reward function r is the intended reward function by the trainer that is learned using SABL.

We develop three mathematical models (linear, sigmoidal, and threshold) to map each computed entropy to the agent's action execution speed,

$$T = \theta_f + (\theta_s - \theta_f) * H, \qquad (3)$$

$$T = \theta_f + \frac{\theta_s - \theta_f}{1 + e^{-10(H - \theta_f)}}, \qquad (4)$$

$$T = \begin{cases} \theta_s & H > 0.1 \\ \theta_f & otherwise \end{cases} \qquad (5)$$

Given the entropy of the action selection H, the corresponding step interval T between two consecutive actions is computed at each time step, which reflects the agent's confidence level of the current action choice. The agent speed varies from θ_f seconds (the fastest speed with highest confidence/minimum entropy) to θ_s seconds (the slowest speed with lowest confidence/maximum entropy) per step.

5. EXPERIMENTS

To study how humans want to train the agent in sequential domains, we developed a user study in which participants train a virtual agent to accomplish pre-specified commands by giving reward and/or punishment.

5.1 Domain

Our domain[2] is a simplified simulated home environment. The domain and user study GUI are shown in Figure 1. The domain consists of four object classes: agent, room, object, and door. The visual representation of the agent is a virtual dog, to fit with our initial motivation of humans being able to teach dogs complex tasks. The agent can deterministically move one unit north, south, east, or west, and push objects by moving into them. The objects are chairs or bags; rooms and objects can be red, yellow, green, blue, and purple. Doors (shown in white in Figure 1) connect two rooms so that the agent can move from one room to another. The possible commands given to the agent include moving to a specified colored room (e.g., "move to the blue room") and taking an object with specified shape and color to a colored room (e.g., "move the blue chair to the purple room"). We provide a fixed set of environments and commands to human participants, limiting the human to providing rewards and punishment feedback.

5.2 Design

In our IRB-approved user study, human participants must first pass a color blind test before starting the experiment since the training task requires the ability to identify different colored objects. Second, users fill out a background survey indicating their age, gender, education, history with dog ownership, dog training experiences, and the dog training techniques they are familiar with. Third, users are taken through a tutorial that explains how to interactively reward and punish the virtual dog based on its behavior. The user is told that punishment can be treated as a signal that the virtual dog should consider a different task than the one it was executing. Fourth, users are tested in the same environment as the tutorial to verify that they understood the interface. Participants that failed this basic check for comprehension (11.3%) were excluded from results.

Following the tutorial and verification test, users are requested to train the virtual dog in a series of three environments shown in Figure 2. Each environment has a different level of complexity and is presented to users in a random order. Following training in the three environments, the users are asked to repeat training with a new virtual dog that behaves differently in the same three environments. Upon finishing the second sequence of tasks, users are asked to describe the strategy used when training the agent. Then, we elicit users' preference with respect to constant or adaptive speed of the virtual dog, and ask them to explain how the adaptive speed virtual dog changes its speed. Finally, participants have the option to provide any additional comments about the experiment.

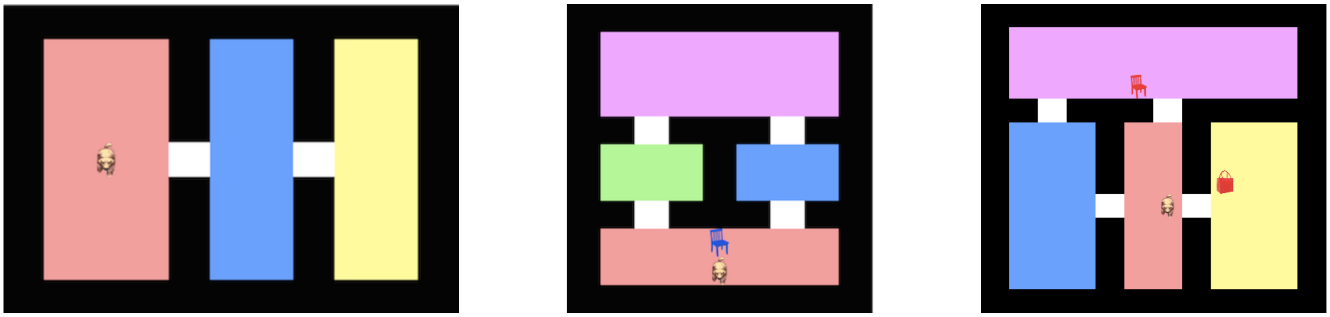
5.3 Preliminary Studies

To implement the novel training condition with adaptive speed agent, we did a set of preliminary studies to determine the threshold of the agent speed ($[\theta_f, \theta_s]$) and the model to be used in the adaptive case.

We first tested four different constant agent speeds, to gain some preliminary insights into the effect of an agent's speed on trainer feedback. The different values of the speed were 0.5, 0.75, 1.0, and 2.0 (seconds per step). It was inspired by

[1] If there are multiple optimal actions for the reward function, then $\pi(s, a|r)$ either arbitrarily selects only one, or is defined to be uniform over the subset of optimal actions.

[2] The video showing the dog training task in our user study is available from `https://www.youtube.com/watch?v=AJQSGD_XPrk`



(a) "Move to the yellow room"

(b) "Move the blue chair to the purple room"

(c) "Move the red bag to the blue room"

Figure 2: The three training environments and their corresponding commands.

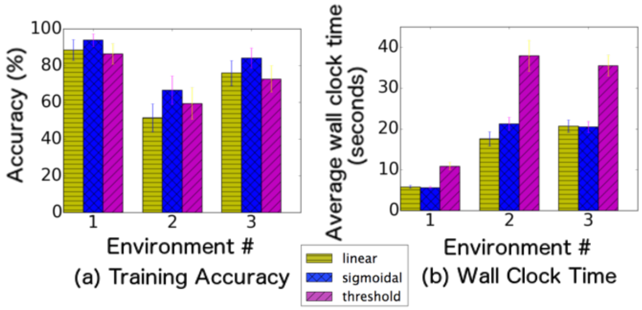


Figure 3: (a) Training accuracy (b) Average wall clock time spent to successfully train the agent when agent's speed changed according to linear, threshold or adaptive model.

the cognitive process study of Hockley [5], which shows that the mean human response time varies from 540.3 $\sim$ 1093.5 ms. Thus, the fastest action execution speed of the agent was assigned as 0.5 seconds/step so that the agent could complete the task with minimal wall clock time while human trainers generally had sufficient time to provide feedback to the virtual dog. Similarly, the slowest action execution speed was set to 2.0 seconds/step to allow each user to have time to think about the strategy and to deliver feedback. We considered the results from 60 users from Amazon's Mechanical Turk and compared the results across different agent speeds. Each participant experienced two agents that moved at different constant fast or slow intervals. There were 30 unique users for each of the four training conditions. Our preliminary analysis suggested that the agent used more wall clock time but fewer actions to learn the appropriate policy when it moved more slowly, receiving more explicit feedback from human trainers. This suggested the agent's speed did affect the trainer feedback and performance. Thus, the threshold of the adaptive agent speed was set as $[0.5, 2.0]$, where the fastest speed θ_f was 0.5 seconds/step, and the slowest speed θ_s was 2.0 seconds/step.

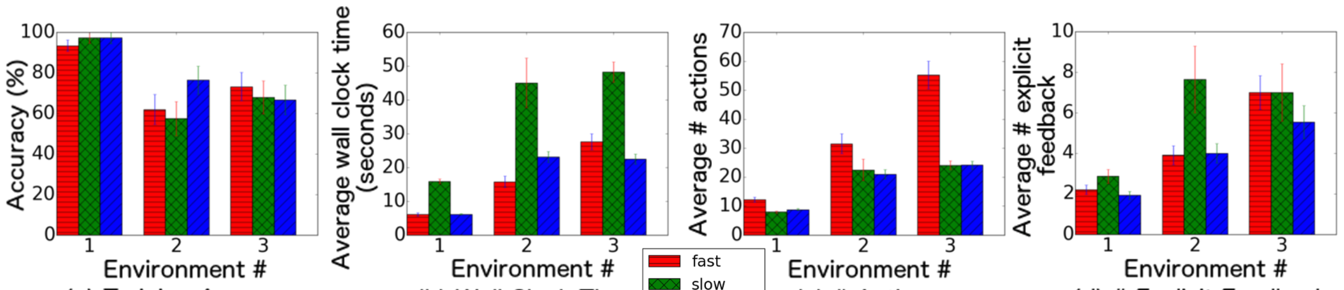
Second, to study the most effective model in modulating the agent speed in adaptive case, we collected data from another 45 Mechanical Turk participants, where each participant was required to train two adaptive speed agents using two of the three speed models (linear, sigmoidal, and threshold). There were 30 participants for each of the three adaptive agents with different speed models. Our preliminary results (shown in Figure 3 (b)) suggested that users spent the most wall clock time to successfully train the adaptive speed agent using the threshold model (Equation 5). This is not surprising since the agent always moved at the slowest interval (2.0 seconds/step) when it had high entropy (low confidence). However, in the other two models, the agent would only move at the slowest interval if it achieved the maximum entropy. A two-way ANOVA test shows that the wall clock time spent training the agent using each of the three adaptive models was significantly different ($p \ll 0.01$). The effect of task complexity on wall clock time used was also statistically significant ($p \ll 0.01$). There was a statistically significant interaction ($p = 0.01$) between the effects of task complexity and adaptive model on wall clock time used. There was statistically significant ($p \ll 0.01$) pairwise difference between the threshold model and linear or sigmoidal model. Compared to the adaptive speed agent using linear model (Equation 3), more users could successfully train the agent using the sigmoidal model (Equation 4) to learn the (near-)optimal policy (shown in Figure 3 (a)). Therefore, we picked the most effective sigmoidal model to map the agent's confidence level to different action execution speeds.

5.4 Final Setup

Based on our preliminary results, we consider two fixed speed agents, one constantly moving fast (0.5 seconds/step) or slow (2.0 seconds/step), and an adaptive speed agent which adjusted its speed (0.5 $\sim$ 2.0 seconds/step) using the sigmoidal model to reflect its confidence level in the current action choice.[3] In the adaptive case, the agent would move quickly when it was confident (had low uncertainty) in its current action choice and slower when it was unsure. Note that action certainty is often different than the agent's confidence in which task and policy is the training target—an agent may have very high confidence in an action if it is in a long hallway and all hypothesized task policies suggest that it should exit the hallway in the same direction.

To investigate whether users can interpret the behavior of the adaptive speed agent correctly, we added another training condition with instructions about when and how the

[3] We note that we have no evidence for or against trainers interpreting the agent's speed as confidence or confusion; however, regardless of interpretation, the speed is computed to reflect confidence and the data supports the claim that it is effective at getting trainers to adjust their feedback.



(a) Training Accuracy (b) Wall Clock Time (c) # Actions (d) # Explicit Feedback

Figure 4: (a) Training accuracy (b) Average wall clock time spent to successfully train the agent (c) Average # actions executed to learn the (near-)optimal policy (d) Average # explicit feedback given when agent's speed was constant fast, constant slow, or adaptive (without instructions), given different environments.

virtual dog changes its speed in the adaptive case—in the tutorial, we explicitly told users that the virtual dog would change its behavior based on how confident it was in its action choice.

In summary, we consider four experimental conditions according to the speed of the agent described as below:

- **Constant Fast Condition**: the agent moves constantly fast (0.5 seconds/step).
- **Constant Slow Condition**: the agent moves constantly slow (2.0 seconds/step).
- **Adaptive Condition without Instructions**: the agent moves adaptively (0.5 $\sim$ 2.0 seconds/step) based on its confidence level without instructions about its changing behavior.
- **Adaptive Condition with Instructions**: the agent moves adaptively (0.5 $\sim$ 2.0 seconds/step) based on its confidence level with instructions about when and how it changes its speed.

6. RESULT ANALYSIS

This section summarizes our user-study results. The user study was published on Amazon Mechanical Turk as a set of Human Intelligence Tasks. We consider data from the 60 unique workers who successfully trained the verification task after the tutorial, showing that they understood the basics of the interface. Each participant trained two virtual dogs that behaved differently in two sequences of the three environments. The two sequences were assigned the same three environments in random order, with two different training conditions. The two training conditions were randomly picked from four different combinations of constant and adaptive conditions. There were the same number of tasks for each of the four training conditions (constant fast, constant slow, adaptive without instructions, and adaptive with instructions).

6.1 User Performance

In this section, we compare the results of the two fixed speed agents (constant fast or slow) against the adaptive speed agent without instructions (the effect of instructions will be discussed later). In all four evaluation metrics we consider—1) training accuracy, 2) wall clock time, 3) number of actions, and 4) amount of explicit feedback—the adaptive speed agent is either competitive with or outperforms the fixed speed agents.

We first evaluated participants' training accuracy by computing the ratio of tasks that were successfully trained to the total number of tasks that were trained by each user. Figure 4(a) shows the average training accuracy achieved in different training conditions given different environments. As we expected, compared against the constant fast or slow agent speed, the training accuracy was not hurt when the agent moved at an adaptive speed. A two-way ANOVA test indicates that the accuracy differences between three training conditions were not statistically significant ($p > 0.05$). However, the accuracy differences between three environments were statistically significant ($p \ll 0.01$), demonstrating that the task complexity did affect the performance of participants in training the agent. We find that environment 1 achieved the highest accuracy since it was easiest, as there were no objects in the environment—the agent only needed to learn the words corresponding to room colors. Contrary to our expectations, the third environment was not harder than the second, even though two objects (a red bag and a red chair) need to be identified in the third environment but only one object (a blue chair) needs to be identified in environment 2. There was no statistically significant ($p > 0.05$) pairwise difference between the second environment and third environment.

Figure 4(b) summarizes the average wall clock time people spent to teach an agent the (near-)optimal policy the first time.[4] Users spent less wall clock time training in the adaptive case than the constant slow agent case, as expected. Wall clock time was minimized in the constant fast agent case in first two environments. It is surprising that the adaptive agent took less wall clock time than the constant fast agent in environment 3, even though the adaptive agent moved slowly when it lacked confidence in its action choice, while the constant fast agent always moved fast in the entire training process. A two-way ANOVA test shows that the wall clock time spent in each of the three training conditions was significantly different ($p \ll 0.01$). The effect of task complexity on wall clock time used was also statistically significant ($p \ll 0.01$). There was a statistically significant interaction ($p = 0.009$) between the effects of task complexity and training condition on wall clock time used. There

[4] Users can decide when to finish training. To standardize timing, we record the time at which the agent first discovers an optimal policy.

was no statistically significant ($p > 0.05$) pairwise difference between the adaptive agent and constant fast agent.

These results show that an agent could accomplish a task using less wall clock time with faster speed. However, it might result in a larger number of actions executed by the agent. Increasing the number of actions executed increases the probability of the agent executing incorrect actions and causing any unwanted effects of those actions. Considering the average number of actions executed by the agent to successfully learn the (near-)optimal policy, shown in Figure 4(c), we find that the adaptive speed agent took fewer actions than the constant fast agent. A two-way ANOVA test shows that the differences of the number of actions executed between the three training conditions or three environments were both statistically significant ($p \ll 0.01$). The interaction effects of task complexity and training condition on number of actions executed were statistically significant ($p \ll 0.01$). There was no statistically significant ($p > 0.05$) pairwise difference between the adaptive speed agent and constant slow agent on this metric.

Combining the first three evaluation metrics, we find that the adaptive speed agent can use 1) significantly less wall clock time with similar number of actions (compared to the constant slow agent) and 2) significantly fewer actions but not significantly worse wall clock time (compared to the constant fast agent) to train the agent, 3) without significantly sacrificing training accuracy.

Lastly, we measured the average amount of explicit feedback given to the agent when it behaved differently. Previous work has shown that the amount of human feedback will decrease over time and forcing a learner to make mistakes can encourage more human feedback [11]. Because fewer actions were used in the adaptive speed case (relative to the constant fast case), we were worried that the agent may receive less explicit human feedback and thus make the problem of inferring the correct task harder for the agent by relying on implicit feedback. As is clear from Figure 4(d), even though fewer actions were used by the adaptive speed agent, roughly the same amount of explicit feedback was given to the constant fast and adaptive speed agents. Learning in fewer actions does not necessarily result in less explicit human feedback. A two-way ANOVA test shows that the amount of explicit feedback delivered in each of the three training conditions or three environments were both significantly different ($p < 0.05$). There was no statistically significant ($p > 0.05$) pairwise difference between the adaptive agent and constant fast agent.

To better understand whether the adaptive speed agent is either competitive with, or outperforms, the fixed speed agents on all four criteria, Figure 5 summarizes the performance of the three agents on the four metrics in Environment 3. Bars show the percentage difference, where the best value for a given metric is 100%. The other bars are computed as $\left(1 - \frac{V_c - V_o}{V_o}\right) \times 100\%$ if the smallest value is the optimum or $\left(1 - \frac{V_o - V_c}{V_o}\right) \times 100\%$ if the largest value is the optimum. V_o is the optimal value of an evaluation metric, and V_c is any other value of the same criterion. We find that the adaptive speed agent outperforms the constant speed agents on wall clock time used and number of actions executed, while maintaining similar training accuracy.

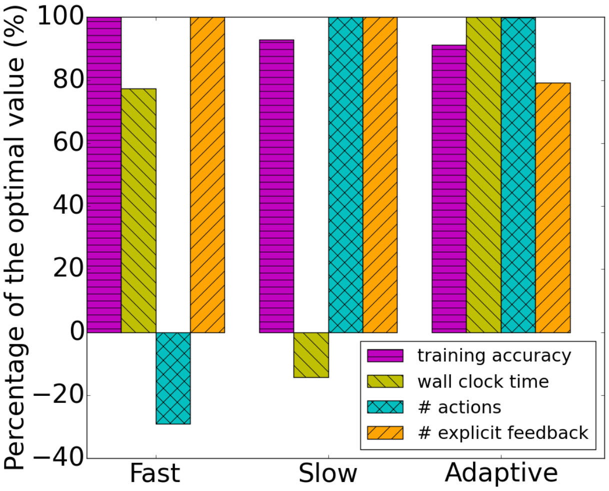


Figure 5: A bar plot showing the percentage of the optimal value each of the four criteria reached given different agent speeds in environment 3. The best result found for a given metric is 100%. The negative value indicates the percentage difference to the optimum is more than 100%. The adaptive speed agent is either competitive with or outperforms the fixed speed agents.

6.2 Speed Feedback Behavior

Recall that one motivation of our implementation of the adaptive speed agent is to encourage users to provide more explicit feedback to the agent when it has higher uncertainty in its action choice. Figure 6 shows the ratio of explicit feedback given to the agent when it moved fast or slow in the adaptive training condition, either with or without instructions, given different environments. As we expected, users tended to give more explicit feedback to the agent when it moved slowly rather than quickly. Considering that the user may provide rewards with certain delay, the slow agent speed provides more time for the trainer to think about the strategy and deliver feedback. It is also possible that the fast agent speed makes the user more nervous and she may hesitate to provide evaluative feedback. For example, one of our subjects reported that "I feel less stressful when the dog slows down and it gives me time to decide whether to reward or punish." A two-way ANOVA test shows that the ratio of explicit feedback delivered by participants when the adaptive speed agent (with or without instructions) moved fast or slowly were significantly ($p \ll 0.01$) different. There was a statistically significant interaction ($p < 0.05$) between the effects of task complexity and agent speed on ratio of explicit feedback given.

To investigate whether users can interpret the behavior of the adaptive speed agent correctly, we analyzed the conditions where they were provided explicit instructions about when and how the virtual dog changed its speed in the adaptive case. In cases where there explanation was provided, we explicitly told users that the virtual dog would change its behavior based on how confident it was in its action choice. Surprisingly, the adaptive speed agent with instructions performed fairly similar with the one without instructions on first three evaluation metrics (training accuracy, wall clock

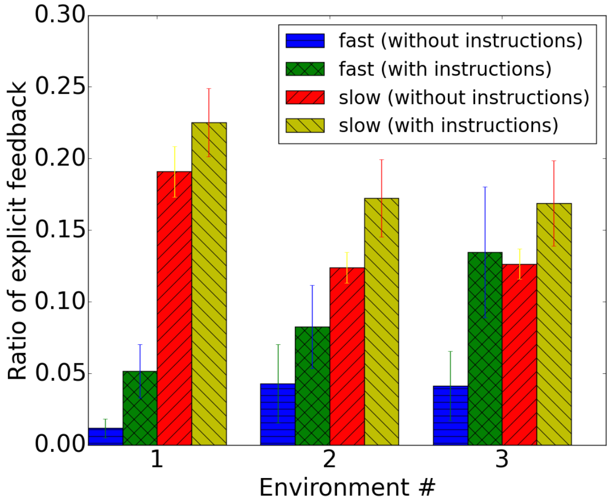


Figure 6: A bar plot showing the ratio of explicit feedback given to the agent when it moved fast or slowly in adaptive training condition with or without instructions, given different environments.

time, and number of actions) we consider before. However, it is worth noting that users provided more explicit feedback in the adaptive case with instructions. The one-way ANOVA test indicates that the number of explicit feedbacks given between the adaptive speed agent with or without instructions was significantly different ($p < 0.05$). Based on the observation that more users interpreted the changing behavior of the agent correctly in adaptive case with instructions (after analyzing users' thoughts about the reason that the agent changed its speed), it is possible that understanding when and how the agent changed its speed did help users to be actively involved in the training and provide more explicit feedback. Future work will investigate our hypothesis.

6.3 User Preference

In this section, to better understand how users interpret the changing speed of the agent, we focus on analyzing the users' responses about constant or adaptive speed agents and consider the results of users from all four training conditions (constant fast, constant slow, adaptive without instructions, and adaptive with instructions) based on different grouping criterion.

Recall that each user interacted with a constant speed agent and an adaptive speed agent in the user study; we first analyze each user's preferences regarding the two different speeds. 86.7% of users expressed a preference. Of these, 73% preferred a constant speed agent since they thought it was easier to predict or control the agent's behavior. However, it is worth noting that among those 27% of users who preferred adaptive speed agent, many of them pointed out that the agent was responsive and behaved like a real dog, giving them insights into its level of comprehension of the task.

Then, we grouped the users based on whether they experienced the constant fast or slow agent, or if they interpreted the behavior of the adaptive speed agent correctly. It is interesting that participants were more likely to prefer the adaptive speed agent if they experienced the constant fast speed agent rather than the constant slow speed agent, or when they understood the changing speed of the agent. Specifically, 36.7% of users expressed that they preferred the adaptive speed agent when they also experienced the constant fast speed agent, while only 13.3% of users preferred the adaptive case when also seeing the constant slow speed agent. Among those users who interpreted the changing speed of the agent correctly, 40% of them preferred the adaptive case. However, the percentage of participants who chose the adaptive case when they misunderstood the agent's behavior was only 22%. Fisher's exact test shows that the differences in user preference for the adaptive speed agent between these two groups were both not statistically significant ($p = 0.07$ and $p = 0.25$ respectively). This suggest an important direction for future work—when, if ever, should an agent behave to maximize performance metrics, even if doing so would not maximize user satisfaction?

7. CONCLUSIONS AND FUTURE WORK

This work considers agents that are better able to learn from human feedback by adapting their speed based on their confidence. Our experimental results show an adaptive speed agent dominates fixed speed agents in four evaluation metrics (training accuracy, wall clock time, number of actions, and amount of explicit feedback). Specifically, an adaptive action execution speed motivates non-expert users to be actively involved in the training and to effectively train agents to perform new tasks in reasonable training times and steps, without sacrificing training accuracy. We demonstrate that the agent's action execution speed can be successfully modulated to encourage more explicit feedback from a human trainer in parts of the state space where the learner has more uncertainty about how to act.

We acknowledge that there are some limitations of this work. First, the agent is constrained to follow the optimal policy of the most likely goal being trained, preventing it from taking exploratory actions to better understand the intention. Second, we did not measure whether trainers switch their feedback strategy during training, particularly if they are more confident in the agent's understanding of the task (trainer's belief about the agent's belief). If such switches in user strategy occurred, the SABL model we currently use cannot gracefully handle dynamic strategy switches.

Future work will 1) consider how human subjects can construct novel tasks and select their own wording (potentially in different languages) to command goals, 2) explore ways to improve the SABL model to consider dynamic strategy switches, 3) allow users to specify a sequence of tasks they would like to train, rather than being constrained to training pre-defined tasks, and 4) investigate how the adaptive speed of an agent can be used to influence human feedback in physical robotic platforms.

Acknowledgements

This research has taken place in part at the Intelligent Robot Learning (IRL) Lab, Washington State University and the CIIGAR Lab at North Carolina State University. IRL research is supported in part by grants AFRL FA8750-14-1-0069, AFRL FA8750-14-1-0070, NSF IIS-1149917, NSF IIS-1319412, USDA 2014-67021-22174, and a Google Research Award. CIIGAR research is supported in part by NSF grant IIS-1319305.

REFERENCES

[1] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85, 1990.

[2] T. Cederborg, I. Grover, C. L. Isbell, and A. L. Thomaz. Policy shaping with human teachers. In *Proc. of IJCAI*, 2015.

[3] S. Griffith, K. Subramanian, J. Scholz, C. Isbell, and A. L. Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2625–2633, 2013.

[4] M. K. Ho, M. L. Littman, F. Cushman, and J. L. Austerweil. Teaching with rewards and punishments: Reinforcement or communication? In *Proc. of COGSCI*, 2015.

[5] W. E. Hockley. Analysis of response time distributions in the study of cognitive processes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 10(4):598, 1984.

[6] C. Isbell, C. R. Shelton, M. Kearns, S. Singh, and P. Stone. A social reinforcement learning agent. In *Proceedings of the fifth international conference on Autonomous agents*, pages 377–384. ACM, 2001.

[7] K. Judah, S. Roy, A. Fern, and T. G. Dietterich. Reinforcement learning via practice and critique advice. 2010.

[8] W. B. Knox, B. D. Glass, B. C. Love, W. T. Maddox, and P. Stone. How humans teach agents - a new experimental perspective. *I. J. Social Robotics*, 4(4):409–421, 2012.

[9] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16. ACM, 2009.

[10] W. B. Knox, P. Stone, and C. Breazeal. Training a robot via human feedback: A case study. In *Social Robotics*, pages 460–470. Springer, 2013.

[11] G. Li, H. Hung, S. Whiteson, and W. B. Knox. Using informative behavior to increase engagement in the TAMER framework. In *Proc. of AAMAS*, 2013.

[12] R. Loftin, B. Peng, J. MacGlashan, M. L. Littman, M. E. Taylor, J. Huang, and D. L. Roberts. Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Journal of Autonomous Agents and Multi-Agent Systems*, pages 1–30, 2015.

[13] J. MacGlashan, M. L. Littman, R. Loftin, B. Peng, D. L. Roberts, and M. E. Taylor. Training an agent to ground commands with reward and punishment. In *Proceedings of the AAAI Machine Learning for Interactive Systems Workshop*, 2014.

[14] P. M. Pilarski, M. R. Dawson, T. Degris, F. Fahimi, J. P. Carey, and R. S. Sutton. Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–7. IEEE, 2011.

[15] V. Pradyot Koruplou, S. Manimaran, B. Ravindran, and S. Natarajan. Integrating human instructions and reinforcement learners: An srl approach. In *Proceedings of the UAI workshop on Statistical Relational AI*, 2012.

[16] V. Pradyot Koruplou, M. S. Sivamurugan, and B. Ravindran. Instructing a reinforcement learner. In *Proceedings of the Twenty Fifth Florida AI Research Society Conference*, pages 56–61, 2012.

[17] H. B. Suay and S. Chernova. Effect of human guidance and state space size on interactive reinforcement learning. In *RO-MAN*, pages 1–6. IEEE, 2011.

[18] R. Sutton and A. Barto. *Reinforcement learning: An introduction.* The MIT Press, 1998.

[19] A. C. Tenorio-Gonzalez, E. F. Morales, and L. Villaseñor-Pineda. Dynamic reward shaping: training a robot by voice. In *Advances in Artificial Intelligence–IBERAMIA 2010*, pages 483–492. Springer, 2010.

[20] A. L. Thomaz and C. Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *AAAI*, volume 6, pages 1000–1005, 2006.

[21] A. L. Thomaz and C. Breazeal. Robot learning via socially guided exploration. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, pages 82–87. IEEE, 2007.