Policy Shaping with Supervisory Attention Driven Exploration

Taylor Kessler Faulkner[1], Elaine Schaertl Short[2], and Andrea Lockerd Thomaz[2]

Abstract— **Robots deployed for long periods of time need to be able to explore and learn from their environment. One approach to this problem has been reinforcement learning (RL), in which robots receive rewards from the environment that allow them to choose optimal actions. To speed learning when human supervision is available, interactive reinforcement learning solicits feedback from a human teacher. However, this approach typically assumes that learning takes place under continuous supervision, which is unlikely to hold in long-term scenarios. We propose an extension to a method of interactive reinforcement learning, policy shaping, that takes into account human attention. Our approach enables better performance while unattended by favoring information-gathering actions when attended and actions that have received positive feedback when unattended. We test our approach in both simulation and on a robot, finding that our method learns faster than policy shaping and performs more safely than policy shaping while no one is paying attention to the robot.**

I. INTRODUCTION

Robots deployed in home environments can benefit from long-term interactive learning, which allows humans in the environment to give a robot feedback over an extended period of time. One approach to interactive learning has been interactive reinforcement learning (RL), in which robots receive both rewards from the environment and feedback from humans. The combination of rewards and human feedback enable robots to take into account human preference when selecting between optimal actions. However, interactive RL typically assumes that the observing human is continuously supervising, and thus learning algorithms choose actions and update their models independently of human attention.

In long-term learning, the assumption that a human will be constantly available to give feedback is unlikely to hold. Continuing learning while no one is present can speed up learning, but can also cause unwanted or dangerous robot behavior during periods of inattention. In previous approaches to interactive RL, if no human is available the robot learns from its environment. In long-term deployment scenarios, continuing to explore the environment as usual while no human is observing may not be optimal behavior. For example, consider a robot deployed in a home, learning the necessary motions to put away dishes. If the robot has a good model of putting cups away but is still exploring to find more efficient methods, exploring without a person around to observe and potentially stop the robot is likely to result in broken glass all over the kitchen. A better approach might be for the robot to put away cups in a potentially suboptimal but trusted way when left alone and to only attempt to learn better actions when supervision is available.

We present an algorithm, attention-modified policy shaping (AMPS), that changes behavior depending on the presence of human attention. Our algorithm is based on *policy shaping* [1], an approach to interactive RL in which the human provides feedback on actions rather than directly providing rewards. We define *attention* as the state of a human watching and maintaining awareness of a robot's actions, and consider the ideal case in which the humans attentional state is fully observable. During periods of attention, the robot favors information-gathering actions that allow it to receive feedback about potentially positive states. When unattended, the robot favors actions that have previously received positive feedback during periods of attention. This approach enables the robot to both learn faster in limited-attention scenarios by increasing exploration when supervision is available, and to learn more safely during human inattention by exploiting known "good" actions when in states that humans have previously seen and for which they have provided positive feedback. If there are actions available that a person has approved, the robot will choose from them.

II. RELATED WORK

AMPS is based on policy shaping (PS), but also integrates results from human-robot engagement and curiosity-driven learning. AMPS incorporates human-robot engagement as it enables the robot to adjust learning styles based on the presence of human engagement. This modification is related to curiosity-driven learning, as AMPS performs information-gathering actions rather than reward-exploiting actions while a person is paying attention.

Policy shaping is a technique developed to solve a number of problems with direct reinforcement feedback from humans [1], [2]. Prior work shows that humans are not good at giving direct state values or rewards to robots using reinforcement learning [3]. To address this, Knox's TAMER system explored a variety of ways to combine human and environmental reward signals [4]. So-called policy shaping has been shown in multiple works to be an effective use of human feedback[1], [4]. In PS human feedback is taken as feedback to individual actions rather than an intermediate reward signal to be combined with the environmental rewards. This enables people to have a clearer idea of what their feedback means to the learning algorithm. Although

This material is based upon work supported by the Office of Naval Research award numbers N000141612835 and N000141612785, National Science Foundation award numbers 1564080 and 1724157, and the NSF-GRFP under Grant No. DGE-1610403

[1]Department of Computer Science, University of Texas at Austin, Austin, TX 78705, USA taylor@cs.utexas.edu

[2]Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78705, USA elaine.short@utexas.edu, athomaz@ece.utexas.edu

we use the policy updating method from PS, this prior work in PS assumes that people are paying attention to the robot throughout the task and that the robot will behave in the same way whether a person is attending or not.

Other work in human-robot interaction (HRI) considers human attention (or engagement) to modify robot behavior [5]–[7]. However, this engagement work modifies robot behaviors directly, not changing robot learning styles based on attention. There has also been work that focuses on robot rather than human attention, which is used either to convince people to engage with the robot [8], [9] or to create more natural social interactions in which the robot shares attention with a human [10]–[12]. These works focus on keeping a person's attention or creating natural interactions with a person already paying attention, but do not focus on what to do when no humans are present.

When no person is paying attention, we incorporate information-gathering actions, as in curiosity-driven learning. Curiosity-driven learning, also known as intrinsic motivation, allows learning agents to explore their environment based on maximizing learning and information potential, not just maximizing rewards or values [13]–[16]. Previous work by Oudeyer et al. has combined curiosity-driven learning with human teachers, creating an agent that chooses whether to follow human advice or explore, but this work also assumes that human feedback is always available to the robot [16].

III. ALGORITHM

We developed an algorithm that changes which actions the robot explores depending on a human supervisor's attentional state. This algorithm combines RL and policy shaping.

A. REINFORCEMENT LEARNING

We formulate our task as a Markov Decision Process (MDP), and use Q-Learning, an off-policy reinforcement learning method [17], which learns Q-values for each state and action to solve a Markov Decision Process (MDP). An MDP is defined by (S,A,T,R,γ), where S is a set of states, A is a set of actions, T is a transition probability function $S \times A \rightarrow Pr[S]$, R is a reward function $S \times A \rightarrow \mathbb{R}$, and γ is a discount factor $0 \leq \gamma \leq 1$. RL methods attempt to select a policy $\pi : S \times A \rightarrow \mathbb{R}$ that achieves the maximum expected reward available in the environment. Q-values $Q(s,a)$ estimate the expected future reward when taking action $a \in A$ in state $s \in S$. We use Boltzmann exploration [18], for which the probability of taking each action is

$$Pr_q(a) = \frac{e^{Q(s,a)/\tau}}{\sum_{a'} e^{Q(s,a')/\tau}} \tag{1}$$

We set τ to 0.5, where τ is an exploration constant decreased by 1% each learning episode. The Q-learning parameters α and γ are set to 0.1 and 0.9 respectively to maximize the performance of policy shaping on our chosen problem.

B. POLICY SHAPING

We incorporate policy shaping with Q-learning to add human feedback. Policy shaping incorporates positive and negative feedback from human teachers into policy choices. Inconsistent human feedback is accounted for by a parameter C, where C is the probability that any feedback from the human teacher is correct. In this work, we set $C = 0.9$, estimating that the teacher is correct 90% of the time, a number consistent with if not lower than results seen in our experimental studies in section V. The probability that any action in a state is good is

$$Pr_c(a) = \frac{C^{\delta_{s,a}}}{C^{\delta_{s,a}} + (1-C)^{\delta_{s,a}}}, \tag{2}$$

where $\delta_{s,a}$ is the difference between positive and negative feedback signals that have been received for state s and action a [1]. The final probability of taking any action is

$$Pr(a) = \frac{Pr_q(a)Pr_c(a)}{\sum_{\alpha \in A} Pr_q(\alpha)Pr_c(\alpha)} \quad . \tag{3}$$

as used in [2].

```
while the robot is learning do
    follow Q-Learning
    if person is paying attention then
        with 50% chance, prioritize a ∉ A_seen
        if no available actions in A_seen then
            follow Policy Shaping
        end
        otherwise prioritize a ∈ A_good
        if no available actions in A_good then
            follow Policy Shaping
        end
    else
        prioritize a ∈ A_good
    end
end
```

Algorithm 1: Attention-Modified Policy Shaping

C. ATTENTION-MODIFIED POLICY SHAPING

Our algorithm chooses actions based on the teacher's attention, as shown in Algorithm 1. For each state, the agent keeps track of the actions that the teacher has seen, A_{seen}, and the actions that have received more positive than negative feedback, A_{good}. In this work, when a person is paying attention, the algorithm randomly chooses with 0.5 probability between taking an action that provides new information (the action is not in A_{seen}), and taking an action that might lead to a better part of the state space (the action is in A_{good}). If either A_{good} or A_{seen} is empty when the agent attempts to choose an action from the set, the agent follows the original PS algorithm. When there is no one paying attention, the agent maximizes the predictability of its actions by choosing only from $a \in A_{good}$, following the original PS algorithm if no such action is available. When the agent is choosing from a reduced set of possible actions, AMPS calculates the probabilities of each action using Equation 3 with the reduced set rather than all possible actions.

In this work, periods of attention or inattention are predetermined, not sensed by the robot. The assumption of

perfect attention detection allows us to directly compare our algorithm with policy shaping. In future work, the human's attentional state will be taken from noisy perception, as has been done in prior work [7], [19]–[22].

IV. SIMULATION EXPERIMENT

We compare our algorithm with the prior approach to policy shaping on a simulated cup placement task. The robot's goal is to push a cup to a desired location on a table, without pushing the cup off the table. This task could be used to put away cups on a shelf in specific locations; cups on the edge of a shelf are easier for humans to reach at a later point. The table is represented by a 6 by 8 grid in simulation.

A. EXPERIMENTAL DESIGN

The goal location for the cup, loc_G, is on the edge of the table, at grid square (5,3) with the grid indexed from zero. This task is well-suited to PS because without human feedback, reinforcement learning will avoid the edges of the table during learning since they are near dangerous states. PS allows people to guide the robot towards loc_G to allow faster learning.

We formulate the problem as an MDP with $S = (x, y)$, the location on the table grid, and $A = \{\text{north, south, east, west, end}\}$, where the first four actions represent a push in that direction and "end" finalizes the position of the current cup and generates a new cup on the table. For the transition function T, each action pushes one grid square in the specified direction. The reward is +100 for ending on loc_G, where this reward is given as the robot pushes the cup onto the location and taken away if it is pushed off of the location. There is a penalty of -125 if the cup falls off the table. All other states have a penalty of -1 to encourage quick travel to the goal.

To represent the human teacher, we use an oracle that gives positive feedback when the agent moves towards loc_g and negative feedback when the agent moves away from loc_g. The oracle has two modes: "attentive" and "inattentive". The "inattentive" oracle never gives feedback, while the "attentive" oracle gives feedback 90% of the time, comparable to a human teacher who may not provide complete feedback even when paying attention.

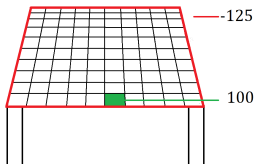


Fig. 1. Example task environment

B. EXPERIMENTS

The agent learns the cup placement task using our algorithm and the prior approach to PS. loc_G and the start location of the cup (2,1) remain the same throughout.

C. RESULTS

Figure 2 shows the learning curves for our algorithm and the prior approach to PS with the oracle paying attention for two sessions of ten episodes. The shaded sections of the background indicate attention from the oracle. The proposed approach performs comparably to AMPS during the first round of attention, but strongly outperforms the prior approach during the period of inattention that follows. In subsequent episodes without attention, performance is greatly improved. The average area under the AMPS reward curve (Mean $(M) = 7024.025$, Standard Deviation $(SD) = 548.566$) is 44% greater than the average area under the PS reward curve ($M = 4877.61, SD = 1357.4$), $t(198) = 14.587, p < 0.05$ (using Welch's t-test). These results suggests that our approach is learning good actions to take during attention by exploring the environment and exploiting the oracle's feedback, allowing the performance while unattended to be safer.

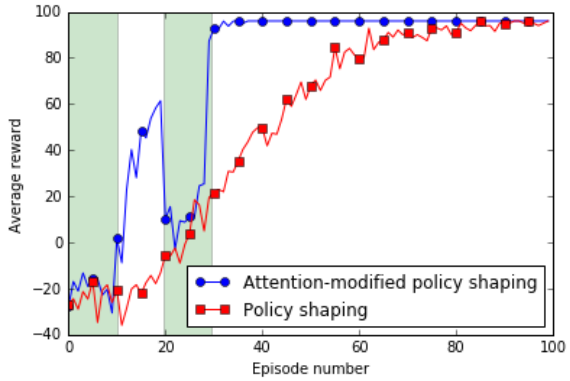
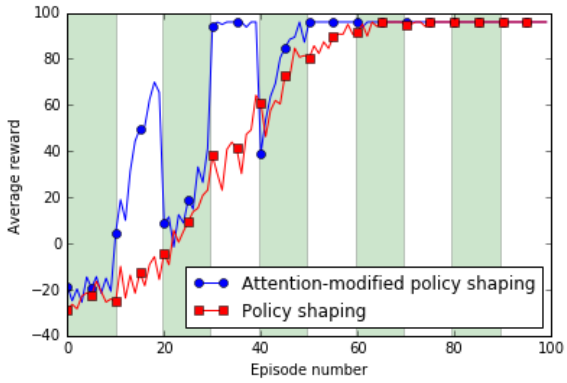


Fig. 2. Total rewards during learning for 100 episodes. All rewards are averaged over 100 runs. The shaded background indicates attention.

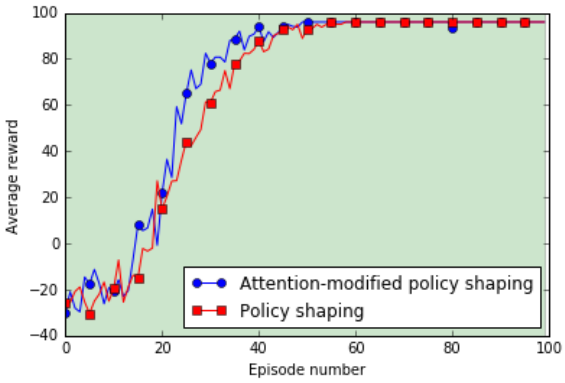
Figure 3 shows the result of adding more attention from the oracle throughout the learning process. The difference between the AMPS and PS learning curves decreases as more attention is added, as PS is able to learn more quickly by receiving more feedback. When the oracle pays attention 50% of the time, the percent increase between the average area of PS ($M = 5441.34, SD = 775.347$) and AMPS ($M = 6853.575, SD = 679.433$) is 25.95%, $t(198) = 13.63$, $p < 0.05$ (using Welch's t-test). When the oracle pays attention for the entire learning process, the percent increase between the area of PS ($M = 6445.975, SD = 546.658$) and AMPS ($M = 6832.095, SD = 404.731$) is 5.99%, $t(198) = 5.648$, $p < 0.05$. With more aggressive exploration, PS could potentially achieve the same average rewards as AMPS during constant attention. However, in addition to faster learning under intermittent attention, the benefit of AMPS is that while this method explores during periods of attention, it falls back to exploitation of human feedback while no one is paying attention, which enables safer performance.

V. REAL-WORLD EXPERIMENT

We also tested our algorithm with naive users supervising a robot performing the cup-pushing task in the real world. The robot pushed a cup on a table divided into a 6 by 8



(a) Oracle pays attention 50% of the time.



(b) Oracle pays attention throughout learning process.

Fig. 3. Total rewards during learning for 100 episodes. All rewards are averaged over 100 runs. The shaded background indicates attention.

grid, with a goal location on the edge of the table. Refer to the video attachment to see an example of this experiment. Based on our simulation results, we hypothesize that AMPS will achieve higher rewards during periods of inattention and a greater total reward over all episodes than PS.

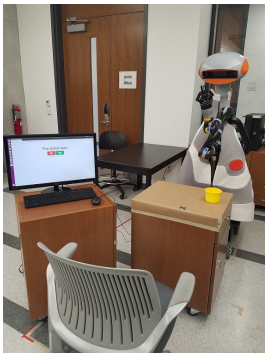


Fig. 4. Robot used during experiments.

A. Robot Setup

We used a robot, Poli, with a Kinova JACO arm with 7 degrees of freedom and a Robotiq 2-finger adaptive gripper, shown in Figure 4. To push the cup, the robot placed its closed gripper inside the cup and moved it a predetermined distance forward, backward, right, or left. The state of the cup was calculated by the position of the gripper over the table by determining in which grid square the robot's gripper location falls. The table was always placed in the same location in front of the robot.

The robot stated the direction in which it planned to move the cup before attempting the move. During the task, if the robot tried to push the cup in a direction but failed due to the cup catching on the table or a manipulator malfunction, or the cup fell off the table, we moved the cup to where the robot expected it to be given the robot's statement. If the robot arm caused an error that stopped the learning process, we restarted learning from the last saved episode. This only happened once during the experiments, on a round of inattention. To control the length of the study, we capped the number of moves per learning episode to twenty pushes. If twenty pushes were reached, the robot asked for the cup to be placed back at the start position.

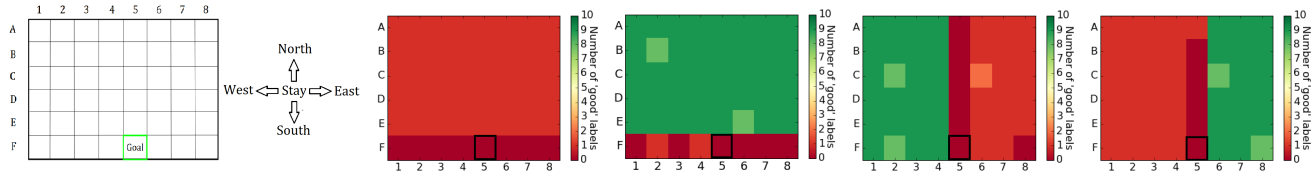
B. EXPERIMENTAL DESIGN

We marked the goal and start locations for the cup on a tabletop, without explicitly marking the grid. An interface was provided with a "Bad" and a "Good" button that could be clicked to send positive or negative feedback to the robot. After taking an action, the robot waited for a response and assumed that no response is given after a timeout. We brought in participants from the campus community to observe the robot and provide feedback while our robot learned the cup pushing task. Each participant observed either the AMPS or PS algorithm. We asked people to click the "Bad" button if they thought an action was bad and the "Good" button if they thought an action was good, paying attention only to the direction of the most recent push action.

Participants gave feedback for the first ten episodes, ignored the robot for five episodes, came back to give feedback for another four episodes, and let the robot learn on its own for one more episode. During the periods of inattention, participants were asked to sit behind a curtain out of view of the robot, and complete a survey designed to capture how they were making decisions about feedback. Each participant looked at an image of a grid with a goal state highlighted in green, see Figure 5(a). For all 48 grid squares in randomized order, we asked them to say whether each action choice (north, east, south, west, and stay) from that square was a "good," "bad," or "neutral" action. After four participants, two of which were used in our data analysis, we noted that there was occasional directional confusion, so we made the instructions more clear by explicitly listing the grid square the cup would be in before and after the action. In Figure 5 b-e, we show a heat map of the participants' responses, where red indicates a low number of "good" markings and green indicates a high number of "good" markings.

C. RESULTS

Figure 6(b) shows the rewards for each episode over all participants. Fourteen participants came in for the study, and four participants were dropped due to robot or human error. Figure 6(a) shows the average rewards for each episode over all participants. To find the average rewards for episodes twenty-one through one hundred and fifty, we save the robot's learning progress after each participant leaves, and then finish learning in simulation using the previously



(a) Participant survey (b) North (c) South (d) East (e) West

Fig. 5. Participant survey feedback for robot actions, with goal circled in black.

described simulation environment. We run the simulation one hundred times for each user, which gives us the average performance of both AMPS and PS over multiple trials. Simulating this process multiple times allows us to show how our algorithm will be expected to perform on average.

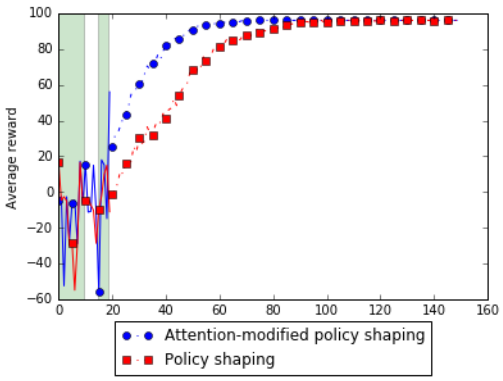
The average area under the AMPS learning curve during the time that the participant was in the lab (the first twenty episodes) ($M = -127.4, SD = 348.566$) is slightly higher than the average area under the PS curve during the first twenty episodes. ($M = -180.9, SD = 141.216$), $t(8) = 0.285$, $p = 0.787$. The average area under the AMPS simulated learning curve from episodes 20-150 ($M = 11491.252, SD = 818.651$) is higher than the average area under the PS simulated learning curve ($M = 10103.123, SD = 1130.163$), $t(8) = 1.989$, $p = 0.085$. Figure 6(b) shows that there is significant noise in the learning progress of the agent during the first twenty episodes, caused by random factors in RL that cause variation in the rewards received early in the learning process. However, an improvement can still be seen during the second period of inattention. The area under the simulated AMPS learning curve also has a lower variance than that of the simulated PS learning curve.

In Figure 7, we see that algorithm performance for both AMPS and PS varies with amount of feedback given per user. The amount of feedback ranges from 47 to 88. The participants' feedback to the robot during the experiments closely matched the feedback of the oracle used in simulation, in which feedback was positive if the cup moved towards the goal location and negative if it moved away from the goal location. The survey responses suggest that the simulation results are indicative of the performance of the simulator with a human oracle (see Figure 5). Two participants did not give feedback for state F8.

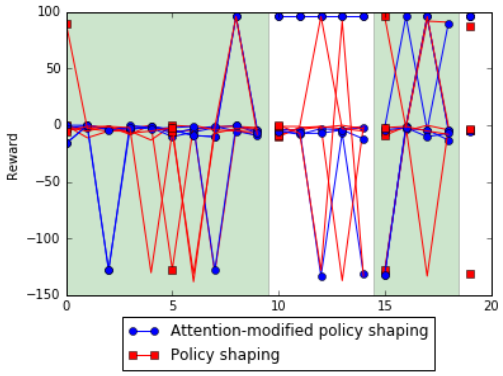
VI. DISCUSSION

Our results suggest that the average area under the AMPS learning curve is consistently higher than the average area under the PS learning curve. Therefore, after the person stops paying attention to the robot and leaves the room, the robot can be expected to perform better on average using AMPS over PS. The lower variance in the average area under the AMPS curve may allow more trust in the learning algorithm overall, as it provides more consistent performance.

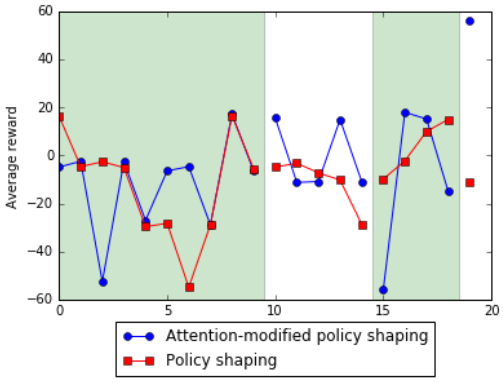
In Figure 6(b), the two algorithms perform similarly during both attention and inattention. We would only expect AMPS to outperform PS on average during inattention during early rounds, as shown in simulation, and the first period of inattention is short. The difference between the two



(a) Results averaged over all participants. The dashed line represents simulated results. The first twenty episodes are completed during the human study.



(b) The first twenty episodes of Figure 6(a), with each participant's data shown over 20 episodes.



(c) The first twenty episodes of Figure 6(a), with average values shown.

Fig. 6. Total rewards during learning for 150 episodes. The shaded background indicates attention.

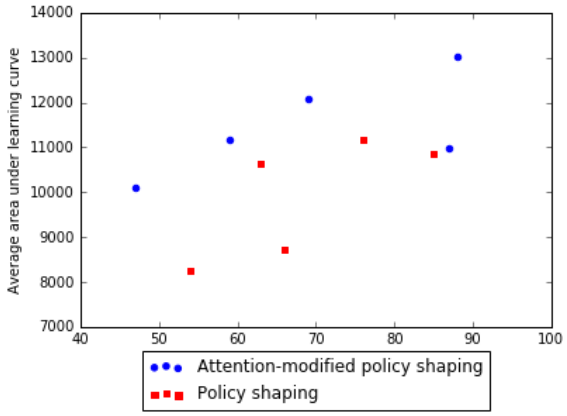


Fig. 7. Amount of feedback given by participants and the resulting areas under the learning curves

algorithms can be seen in Figure 6(a), which shows the second longer period of inattention. Figure 7 suggests we would see better performance from both AMPS and PS given more feedback from users. AMPS may also be sensitive to the amount of feedback given, as the more positive feedback it has received, the longer the robot will be able to act without trying new and unseen actions.

AMPS may more closely match people's expectations of how the learning process should proceed. The robot prioritizes actions that add and confirm task knowledge while the human teacher is present, and prioritizes listening to prior positive feedback while no teacher is present. This behavior is similar to social referencing, which serves a role in human development by allowing infants to explore new actions while looking to a trusted authority for feedback [23].

Future work could add more participants to better see the patterns that emerge on average using AMPS, and test the effect of adding more people with variable feedback accuracy. To ensure that AMPS will perform in real-world contexts, more tasks should be tested in longer-term studies. We plan to extend this work to use more advanced curiosity-driven learning methods during periods of attention, rather than only biasing towards immediate actions that a person has not seen. Large state spaces could benefit from a similarity metric to determine what actions to choose, which would allow the robot to pick from lists of actions that are similar to those to which a person has given attention or positive feedback. This metric may allow robots to choose safer actions even when in previously unexplored areas of the state space.

VII. CONCLUSION

We propose that interactive RL agents should change the way they learn based on human attention. While robots can still learn without attention, AMPS allows robots to take advantage of human attention while attempting to behave more optimally while unattended. Our results suggest that exploring new actions and confirming the performance of positively marked actions while attended and exploiting previously positively marked actions while unattended produces safer and more consistent performance than policy shaping.

REFERENCES

[1] S. Griffith, K. Subramanian, J. Scholz, C. L. Isbell, and A. L. Thomaz, "Policy shaping: Integrating human feedback with reinforcement learning," in *Advances in neural information processing systems*, 2013, pp. 2625–2633.

[2] T. Cederborg, I. Grover, C. L. Isbell, and A. L. Thomaz, "Policy shaping with human teachers." in *IJCAI*, 2015, pp. 3366–3372.

[3] A. L. Thomaz and C. Breazeal, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," *Artificial Intelligence*, vol. 172, no. 6-7, pp. 716–737, 2008.

[4] W. B. Knox and P. Stone, "Tamer: Training an agent manually via evaluative reinforcement," in *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*. IEEE, 2008, pp. 292–297.

[5] P. Rani and N. Sarkar, "Operator engagement detection and robot behavior adaptation in human-robot interaction," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 2051–2056.

[6] Q. Xu, L. Li, and G. Wang, "Designing engagement-aware agents for multiparty conversations," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 2233–2242.

[7] M. P. Michalowski, S. Sabanovic, and R. Simmons, "A spatial model of engagement for a social robot," in *Advanced Motion Control, 2006. 9th IEEE International Workshop on*. IEEE, 2006, pp. 762–767.

[8] A. Bruce, I. Nourbakhsh, and R. Simmons, "The role of expressiveness and attention in human-robot interaction," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4. IEEE, 2002, pp. 4138–4142.

[9] C. L. Sidner, C. Lee, C. D. Kidd, N. Lesh, and C. Rich, "Explorations in engagement for humans and robots," *Artificial Intelligence*, vol. 166, no. 1-2, pp. 140–164, 2005.

[10] M. W. Doniec, G. Sun, and B. Scassellati, "Active learning of joint attention," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*. IEEE, 2006, pp. 34–39.

[11] C.-M. Huang and B. Mutlu, "Robot behavior toolkit: generating effective social behaviors for robots," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 25–32.

[12] M. Staudte and M. W. Crocker, "Visual attention in spoken human-robot interaction," in *Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on*. IEEE, 2009, pp. 77–84.

[13] J. Achiam and S. Sastry, "Surprise-based intrinsic motivation for deep reinforcement learning," *arXiv preprint arXiv:1703.01732*, 2017.

[14] J. Schmidhuber, "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 230–247, 2010.

[15] N. Chentanez, A. G. Barto, and S. P. Singh, "Intrinsically motivated reinforcement learning," in *Advances in neural information processing systems*, 2005, pp. 1281–1288.

[16] P.-Y. Oudeyer *et al.*, "Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner," *Paladyn*, vol. 3, no. 3, pp. 136–146, 2012.

[17] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[18] C. J. Watkins, "Models of delayed reinforcement learning," Ph.D. dissertation, Ph. D. thesis, Cambridge University, 1989.

[19] D. Lala, K. Inoue, P. Milhorat, and T. Kawahara, "Detection of social signals for recognizing engagement in human-robot interaction," *arXiv preprint arXiv:1709.10257*, 2017.

[20] M. E. Foster, A. Gaschler, and M. Giuliani, "Automatically classifying user engagement for dynamic multi-party human–robot interaction," *International Journal of Social Robotics*, vol. 9, no. 5, pp. 659–674, 2017.

[21] J. Sanghvi, G. Castellano, I. Leite, A. Pereira, P. W. McOwan, and A. Paiva, "Automatic analysis of affective postures and body motion to detect engagement with a game companion," in *Proceedings of the 6th international conference on Human-robot interaction*. ACM, 2011, pp. 305–312.

[22] C. Rich, B. Ponsler, A. Holroyd, and C. L. Sidner, "Recognizing engagement in human-robot interaction," in *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*. IEEE, 2010, pp. 375–382.

[23] S. Feinman, "Social referencing in infancy," *Merrill-Palmer Quarterly (1982-)*, pp. 445–470, 1982.