2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Using Learning Curve Predictions to Learn from Incorrect Feedback*

Taylor A. Kessler Faulkner[1] and Andrea L. Thomaz[2]

***Abstract*— Robots can incorporate data from human teachers when learning new tasks. However, this data can often be noisy, which can cause robots to learn slowly or not at all. One method for learning from human teachers is Human-in-the-loop Reinforcement Learning (HRL), which can combine information from both an environmental reward and external feedback from human teachers. However, many HRL methods assume near-perfect information from teachers or must know the skill level of each teacher before starting the learning process. Our algorithm, Classification for Learning Erroneous Assessments using Rewards (CLEAR), is a feedback filter for Reinforcement Learning (RL) algorithms, enabling learning agents to learn from imperfect teachers without prior modeling. CLEAR is able to determine whether human feedback is correct based on observations of the RL learning curve. Our results suggest that CLEAR improves the quality of human feedback — from 57.5% to 65% correct in a human study — and performs more reliably than baselines by matching or outperforming RL without human teachers in all tested cases.**

I. Introduction

Many prior methods in Human-in-the-loop Reinforcement Learning (HRL) work quite well with human feedback but do not fully address human limitations, assuming that the robot has a known prior on how correct the teacher's feedback is [1]. HRL gives robots two sources of information: an environmental reward function and feedback from human teachers. Learning robots can use one of these sources to confirm the performance of the other. To learn from inaccurate teachers, we enable robots to decide which teacher-provided information to trust, using additional sources of information such as the reward function in HRL. This work is motivated by circumstances in which people give consistently bad feedback on some states and consistently good feedback on others, which may happen if teachers are confused about the task goal or how the robot functions.

We present an algorithm, Classification for Learning Erroneous Assessments using Rewards (CLEAR), that uses achieved cumulative rewards to learn whether binary feedback ("good" or "bad" responses to actions) received from a teacher is correct or incorrect over time. This work uses a classifier to store predictions of the slope of the learning curve based on observed state-action pairs and add supplemental binary feedback to the human teacher's feedback. The CLEAR algorithm filters feedback to a learning robot and explores the feedback performance at the beginning of the learning process. We test this algorithm against Policy Shaping [2] and Q-Learning with varying scenarios of human misunderstandings of a robot. The results suggest that using CLEAR as a feedback filter matches or exceeds the performance of Q-Learning over many levels of feedback quality, while the performance of Policy Shaping varies considerably based on feedback quality. This observation supports our claim that CLEAR can help robots learn more dependably than a baseline HRL method.

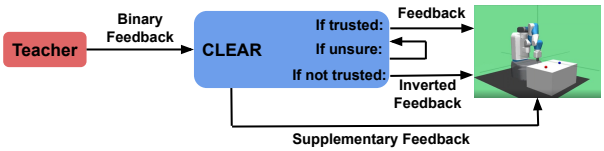


Fig. 1. CLEAR takes input from teachers and filters it for a robot learning using Human-in-the-loop Reinforcement Learning (HRL). Feedback is passed directly through if trusted, inverted if not trusted, and withheld if CLEAR is unsure. CLEAR can also give supplementary feedback over time. The robot shown is the simulated robot environment that we use, created using HIPPO Gym [3], MuJoCo [4], and OpenAI Gym [5].

II. Background

HRL supplements a Markov Decision Process (MDP) framework with guiding information from a human teacher [1]. HRL can take human data in many different forms. Some prior work takes feedback on past actions, whether scalar-valued or binary [6], [7], [2], [8]. Other work takes advice on future actions, which lets the human guide an agent [9], [10], [11], [12], or intervention on current actions into danger [13]. HRL can also learn from full demonstrations from a teacher [14], [8], [15], [16], [17] with the addition of a reward function to speed learning, or it can allow users to compare and rank trajectories demonstrated by a robot [18].

There is also prior work on algorithms that address the possibility of incorrect feedback from users [14]. Some work assigns trust or weight to human feedback, either using a static weight to human input throughout learning [2] or slowly decreasing the weight over time [7]. Kurenkov et al. [19] uses multiple teachers to confirm performance. Sridharan [20] stores multiple policies from a reward function and one policy learned from feedback. This method then compares these policies with each other to calculate trust. Work by Lin et al. [12] learns to estimate trust in the teacher over time but relies on comparisons to the currently

*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]Taylor A. Kessler Faulkner completed this work during her Ph.D. in the Department of Computer Science at The University of Texas at Austin. She is currently with the Department of Computer Science and Engineering, University of Washington. taylorkf@uw.edu.

[2]Andrea L. Thomaz is with the Department of Electrical and Computer Engineering, The University of Texas at Austin. athomaz@ece.utexas.edu

learned Q-values. As Q-values are likely incorrect near the beginning of the learning process, this method may discount good feedback at the beginning. Our prior work [21] also learns whether or not to trust feedback over time with the Revision Estimation from Partially Correct Resources (REPaIR) algorithm. However, this algorithm begins filtering feedback early in the learning process and requires storing cumulative reward values for each observed state-action pair rather than learning the quality of areas of the state space over time. When state-action pairs are unobserved, REPaIR has no data for them and thus cannot provide improved feedback until later in the learning process. Some Inverse Reinforcement Learning (IRL) methods estimate whether input from people is correct [22], [23], [24]. However, these require full demonstrations or external information like the relative occurrence of incorrect demonstrations [23], [24] or trajectory rankings [22].

A. Q-Learning

Q-Learning [25] is an off-policy model-free Reinforcement Learning (RL) algorithm that uses a Markov Decision Process (MDP) to learn the relative values of states and actions. The definition of an MDP is the tuple (S, A, T, R, γ): S is a set of states, A is a set of actions, $T(s, a, s')$ is a transition function giving the probability of transitioning to $s' \in S$ when taking $a \in A$ in $s \in S$, $R(s, a, s')$ is a reward function for the transition from $s \in S$ to $s' \in S$ using $a \in A$, and γ is a discount factor.

Q-learning uses the discount factor γ from the MDP and a learning rate, α. These, along with observed rewards, are used to learn Q-values using a Bellman update over multiple learning episodes, where each episode is a learning trajectory of state-action pairs, where taking the state-action pair (s, a) leads to state s'

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_{a' \in A} Q(s', a') - Q(s,a)].$$

We use Boltzmann exploration with Q-learning [26], [25] to encourage a learning agent to explore new states and actions rather than purely exploiting the currently learned Q-values. Using Boltzmann exploration, the probability of taking any action a in state s is

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

where $Q(s, a)$ are the currently learned Q-values, and τ is a temperature parameter.

B. Policy Shaping

Policy Shaping is a method of HRL that takes binary feedback on state-action pairs from human teachers [27], [2]. We use Policy Shaping as the underlying HRL algorithm for our experiments. Rather than acting as a reward, the received feedback shapes the robot's policy. Thus the reward function and Q-values are unaffected by the feedback. Instead, the exploration of the robot is modified. For policy shaping, the probability of taking each action given feedback is

$$\Pr_c(a|s) = \frac{C^{\Delta_{s,a}}}{C^{\Delta_{s,a}} + (1-C)^{\Delta_{s,a}}}.$$

where $\Delta_{s,a}$ is defined as the difference in positive and negative values given by the teacher to action a in state s, and $C \in [0, 1]$ is a trust parameter, with 0 being complete distrust in the human teacher and 1 being complete trust. Using $\Delta_{s,a}$ rather than the count of positive feedback on (s, a) allows for slightly inconsistent feedback. When $C = 0.5$, PS reduces to RL with no feedback.

The probability of taking any action using the Q-values of the MDP is $\Pr_q(a|s)$. Using $\Pr_q(a|s)$ and $\Pr_c(a|s)$, the probability of taking any action $a \in A$ in state $s \in S$ while learning is

$$\Pr_p(a|s) = \frac{\Pr_q(a|s)\Pr_c(a|s)}{\sum_{\alpha \in A} \Pr_q(\alpha|s)\Pr_c(\alpha|s)}.$$

The Policy Shaping algorithm assumes that the teacher's skill level is known beforehand to set the parameter C. When the C parameter does not match the teacher's skill level, Policy Shaping will underperform Q-learning [21].

III. Methodology

CLEAR uses an online learning classifier, C_{CLEAR}, to predict whether the RL learning curve will rise or fall based on state-action pairs. CLEAR combines this information with the environmental reward function R, which is assumed to be correct, to filter feedback. The predictions from C_{CLEAR} determine whether to keep, invert, or discard feedback, giving output similar to REPaIR. However, CLEAR's method of choosing when to keep, invert, or discard feedback does not require thresholds that need to be set by an expert.

C_{CLEAR} [1] takes in state features and actions and outputs a prediction on whether the RL learning curve will rise, fall, or stay the same after the current trajectory. In general, a rising RL learning curve is a positive result, as the goal is to find the highest-performing policy. Falling RL learning curves, in the absence of local minima, suggest a decrease in performance. CLEAR learns to predict the sign of the slope of the learning curve rather than learning the resulting scalar cumulative reward as is done in REPaIR. This problem would require regression and is a difficult problem to solve in larger and more complex state spaces.

A. Algorithm

CLEAR saves the state action pairs of the current episode's trajectory, $traj_e$, and the trajectory that has received the highest cumulative reward, $traj_m$. The cumulative reward for episode e is $R_e = \sum_{t=0}^{T} r_t$ where T is the total number of time steps in an episode, and r_t is the reward received at each time step. The trajectories are composed of (s_t, a_t) tuples, where s_t is the state at time t, and a is the action taken at time t. The current and maximum cumulative rewards are saved: R_e, R_{max}. CLEAR uses this information to preprocess data

[1] Implemented as the scikit-learn multinomial Naive Bayes classifier [28], training performed by partial_fit()

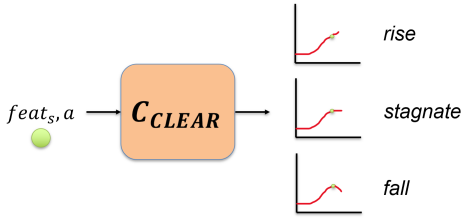


Fig. 2. CLEAR algorithm: this classifier predicts the slope of the RL learning curve based on the current action and state features.

and determine whether the classifier should predict a rising, falling, or stagnating learning curve, as shown in Figure 2. C_{clear} takes as input state features expected to influence the quality of human feedback (e.g., x-y gripper position, joint configuration, etc.) along with the current action. The output is one of three choices: *rise*, *stagnate*, or *fall*.

At the end of each trajectory, CLEAR trains C_{CLEAR}. Each training sample is given with a weight equal to the current episode count squared, weighting samples more heavily as learning continues and observed total rewards are more likely to be incorrect. C_{CLEAR} as follows for each $(s, a) \in traj_e$, where $feat_{s_t}$ is the features of state s_t:

- If $R_e > R_{max}$
$$C_{CLEAR}[(feat_{s_t}, a_t)] = rise$$
- Else if $R_e \leq R_{max} - \frac{current\_episode\_count}{maximum\_episode\_count} * |(R_{max} - R_{min})|$
$$C_{CLEAR}[(feat_{s_t}, a_t)] = fall$$
- Else $C_{CLEAR}[(feat_{s_t}, a_t)] = stagnate$ if $(s_t, a_t) \notin traj_m$

We note that C_{CLEAR} is trained with a "fall" label only if $R_e \leq Rmax - (\text{episode\_count/max\_count}) * |(R_{max} - R_{min}))$, not simply $R_e < R_{max}$. We found this setting to work best in practice, giving a small but widening range to define *stagnation*, as the current episode count gets closer to the maximum number of episodes the experimenter is running.

CLEAR then keeps, discards, inverts, or supplements feedback based on the predicted learning slope. That is, feedback is either directly passed through to the learning algorithm, not passed through at all, inverted by calculating $-\Delta_{s,a}$ as used in Policy Shaping [2], [27], or added by CLEAR. When an HRL algorithm requests feedback, the state-action pairs are given to CLEAR, which then filters feedback by predicting the upcoming slope of the learning curve, $pred_s$. Using the predicted probabilities of C [2], where $probs = C.predict\_proba([s, a])$, $\Pr[pred_s = fall] = probs[fall]$, $\Pr[pred_s = stagnate] = probs[stagnate]$, $\Pr[pred_s = rise] = probs[rise]$. We define the current feedback input as f, and the total feedback received for a state as $\Delta_{s,a}$. Recall that Policy Shaping measures $\Delta_{s,a}$ as the difference in positive and negative feedback, so that a negative $\Delta_{s,a}$ means that the teacher disapproves of the state-action pair, and a positive $\Delta_{s,a}$ means that the teacher approves of the state-action pair. If there is no feedback ($\Delta_{s,a} == 0$), CLEAR supplements feedback f proportional to the probability of the slope direction: $+f$ if $pred_s == rise$, and $-f$ if $pred_s == fall$. In this work, we set $f = \frac{probs[pred]}{2}$, but this setting could be tested further in future work. Specifically, CLEAR returns feedback as follows (assuming feedback is input to a Policy Shaping baseline algorithm [3]):

- If $\Delta_{s,a} < 0$ and $pred_s == fall$, or $\Delta_{s,a} > 0$ and $pred_s == rise$, KEEP
  - Return $\Delta_{s,a}$
- Else if $pred_s = rise$ and $\Delta_{s,a} < 0$, or $pred_s = fall$ and $\Delta_{s,a} > 0$, INVERT
  - Return $-\Delta_{s,a}$
- Else if $\Delta_{s,a} == 0$ and $pred_s == rise$, ADD
  - Return $\frac{probs[rise]}{2}$
- Else if $\Delta_{s,a} == 0$ and $pred_s == fall$, ADD
  - Return $-\frac{probs[fall]}{2}$
- Else, DISCARD
  - Return 0

[2]Predicted using the predict_proba() function [28]

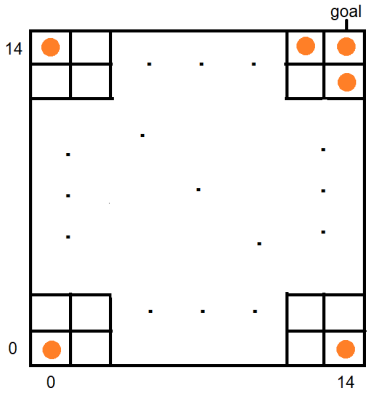
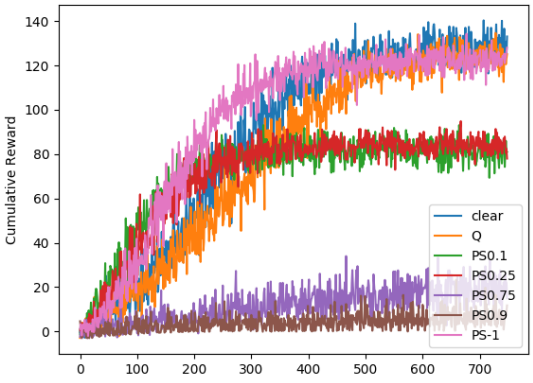


Fig. 3. The state space used for the reaching task, in which the robot must reach the true goal location while ignoring a distractor goal. The placements of the distractor goals are shown in orange for $|S| = 225$

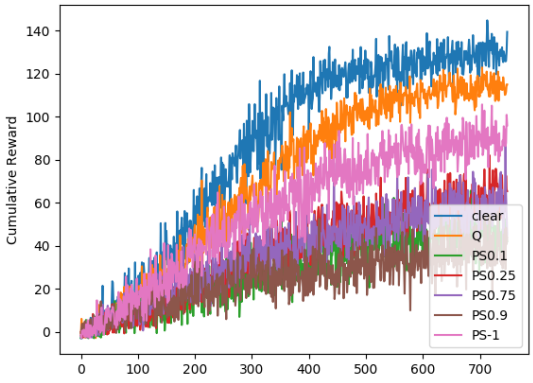
IV. Experiments

For these experiments, we compare CLEAR to baseline RL with no feedback and to Policy Shaping with varying settings of the trust setting $C \in [0, 1]$. C is set to 0.1, 0.25, 0.75, 0.9, and the true percentage of correct feedback $p*$. In the result graphs (Figure 4), PS-1 denotes $C = p*$. For CLEAR, $C = 0.8$, as this trust setting shows a moderate trust in the teacher ($0.5 < C < 1.0$). The baseline algorithms for these experiments use Q-Learning with Boltzmann exploration, with $\alpha, \gamma, \tau = 0.9$. The parameter τ is annealed by multiplying by 0.999 at the end of each episode. Each algorithm is run 100 times for a length of 750 episodes, with each episode ending after $2 * \sqrt{|S|}$, where $|S|$ gives the total number of states.

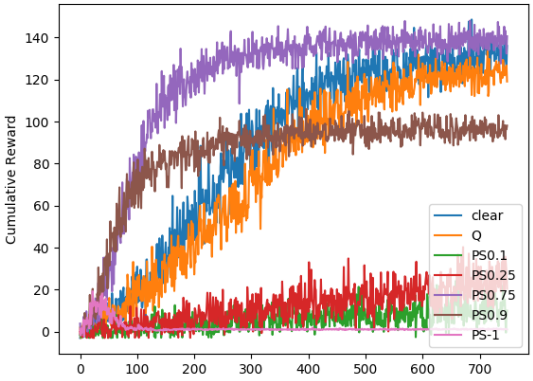
[3]If the baseline algorithm is not Policy Shaping, replace $\Delta_{s,a}$ with the feedback function for the baseline HRL algorithm.



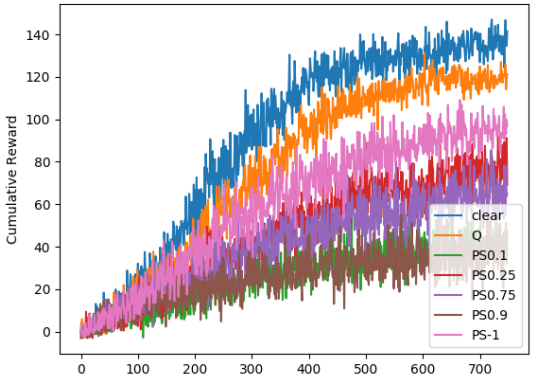
(a) Distractor goal (0,0)



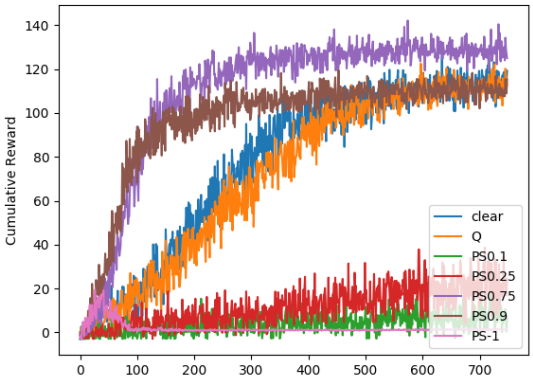
(b) Distractor goal ($|S|-1$,0)



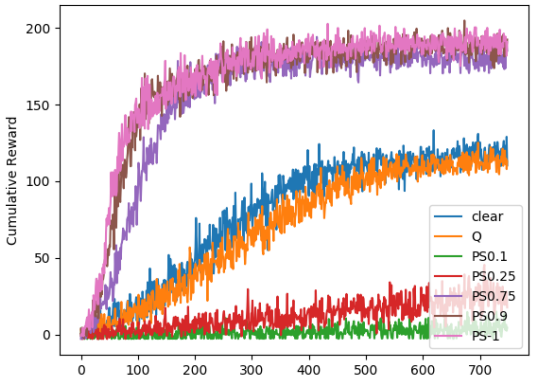
(c) Distractor goal ($|S|-1$,$|S|-2$)



(d) Distractor goal (0,$|S|-1$)



(e) Distractor goal ($|S|-2$,$|S|-1$)



(f) Distractor goal ($|S|-1$,$|S|-1$)

Fig. 4. CLEAR simulation results with varied feedback correctness over 750 learning episodes

A. Simulation

The simulation task is a robot moving its gripper to touch a goal object on a flat surface. The discretized state space is 15x15 ($|S| = 225$, with the true goal at ($\sqrt{|S|}-1, \sqrt{|S|}-1$). The robot arm can take cardinal and diagonal actions, one square at a time, or choose not to move. The robot arm starts at a random location at the beginning of each episode. There is also a distractor goal. In this task, the human confusion pertains to the goal; that is, of two possible goals (the true and distractor), the teacher believes the distractor goal to be the true goal. As shown in Figure 3, the distractor goal is placed at the various locations indicated with an orange dot:

1) (0,0)
2) $(0, \sqrt{|S|} - 1)$
3) $(\sqrt{|S|} - 1, 0)$
4) $(\sqrt{|S|} - 2, \sqrt{|S|} - 1)$
5) $(\sqrt{|S|} - 1, \sqrt{|S|} - 2)$
6) $(\sqrt{|S|} - 1, \sqrt{|S|} - 1)$.

At the beginning of each episode, the robot gripper starts at a random, non-goal, non-distractor-goal state. The goal and distractor goal remain the same over each iteration. We model the simulated human feedback as an oracle that gives perfect feedback to the distractor goal. When the distractor goal is in the same space as the true goal, the simulated human gives perfect feedback to the true goal. The simulated human gives feedback 80% of the time. The sparse reward function is +10 at the true goal, +0.1 at the distractor goal, and -0.1 for all other states. There is a reward at the distractor goal to demonstrate that CLEAR can recover even if there is a local maximum on the human's incorrect goal.

B. Human Study

We ran a human study on Amazon Mechanical Turk with 10 participants. We used a simulated Fetch robot to run a modified version of the FetchReach-v1 task [5] that spans a 6x6 space over a table. This task is equivalent to the reaching task used in the full simulation studies, except that the robot begins at state (0,2) every time rather than a random starting location. The reward function is +10 at the goal state, +0.1 at the distractor state, and -0.1 at all other states. The state features are the x and y coordinates of the robot gripper. The task still requires the robot to reach a goal object with a distractor object present. For this study, the simulated robot had a single true (2,1) and distractor goal object location (2,3). The Amazon Mechanical Turk participants viewed four videos of the robot reaching towards the different objects, taking the following paths (shown in Figure 5):

- Paths to true goal
  - (0,2),(0,1),(1,1),(2,1)
  - (0,2),(1,2),(2,2),(2,1)
- Paths to distractor
  - (0,2),(0,3),(1,3),(2,3)
  - (0,2),(1,2),(2,2),(2,3).

These are example paths that the robot could take to the true goal and the distractor. Since the paths have the same length, the two paths to the goal have an equal reward and the two paths to the distractor have an equal reward. Each participant recorded their feedback for each action.

In order to test CLEAR's performance on human feedback, we began with pretrained classifiers. In order to obtain 100 classifier instances, we ran CLEAR 100 times on the human study task with a simulated teacher giving incorrect feedback to the distractor object. Each CLEAR episode ends after seven actions. We used these pretrained classifiers to classify the human data collected on Amazon Mechanical Turk, in order to determine how well our CLEAR classifier would perform on potentially messy human data.

DISTRACTOR	CLEAR	Q	PS-min	PS-max
(0,0)	65290	59076	3015	70708
$(\|S\| - 1, 0)$	64528	53131	18426	42057
$(\|S\| - 1, \|S\| - 2)$	67682	56605	18970	44423
$(0, \|S\| - 1)$	65626	5780	1376	88052
$(\|S\| - 2, \|S\| - 1)$	57267	53706	1415	481616
$(\|S\| - 1, \|S\| - 1)$	58851	52618	1211	124943

TABLE I

THE MEAN AREA UNDER THE CURVE (AUC) FOR CLEAR, Q-LEARNING (Q), AND MINIMUM AND MAXIMUM PERFORMING POLICY SHAPING (PS-MIN, PS-MAX), GATHERED IN SIMULATION.

V. RESULTS

A. Simulation

The simulation study results are shown in Figure 4 and Table I. We measured the area under the learning curve (AUC) using the composite trapezoidal rule for CLEAR, Q-Learning, the C value that produces the minimum performing PS, and the C value that produces the maximum performing PS, averaged over all 100 iterations. The average AUC shows us the total rewards gather over time on average for each algorithm. Higher AUCs indicate that an algorithm achieved higher total rewards on average. Using a one-way ANOVA and a Tukey posthoc test, CLEAR performs significantly better than Q-Learning ($p < 0.05$) for distractor placements ($|S| - 1$,0) and ($|S| - 1$,$|S| - 2$). However, the PS performance varies by large amounts based on the C parameter. Since the simulated teacher gives feedback based on incorrect knowledge of the robot's true goal state, PS is subject to wide variances in performance without a model of how well the teacher performs. This is likely due to the fact that the parameter C enables the robot to weight incoming feedback against state values appropriately, as long as it is adequately matched to the feedback quality. However, if the C parameter and feedback quality are mismatched, PS will discount correct feedback too much or put too much weight on incorrect feedback.

These results suggest that CLEAR performs more dependably than Policy Shaping with simulated human feedback. However, this does not test its performance with real people and messy feedback. We hypothesize that, even if trained on simulated human feedback, CLEAR could determine what feedback from real human teachers was correct or incorrect.

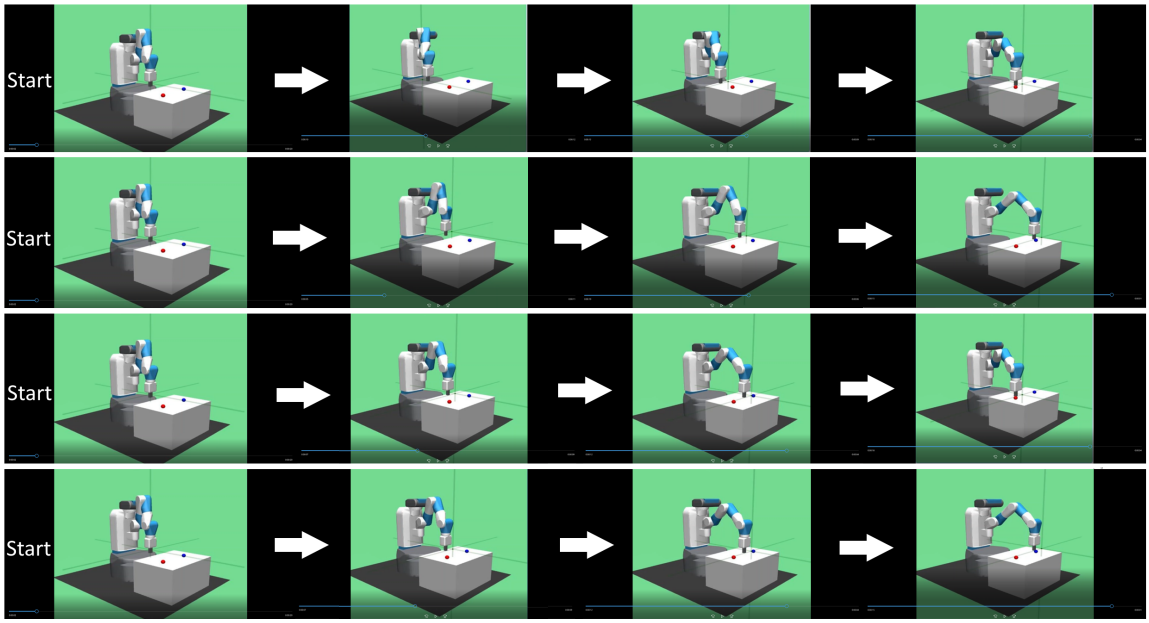


Fig. 5. Videos in simulated robot environment, created using HIPPO Gym [3], MuJoCo [4], and OpenAI Gym [5]. Each row shows a video clip of the robot moving toward one of two goals, using two different trajectories.

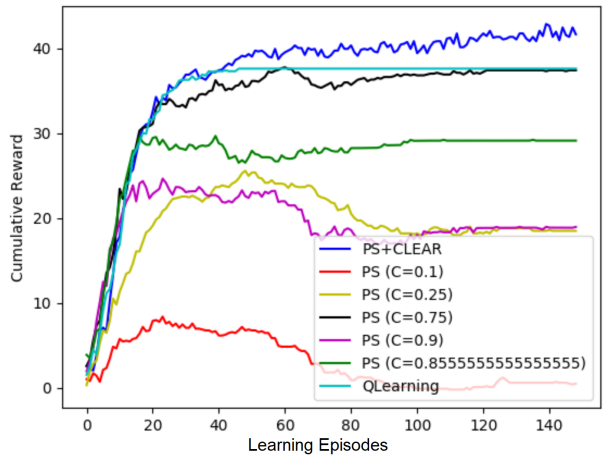


Fig. 6. Simulated performance prior to Amazon Mechanical Turk data.

B. Human Study

Although the participants were instructed to give correct feedback to the blue distractor object, the feedback was not clean, with some participants giving incorrect feedback to both the true and distractor object. Before filtering, the human data was 57.5% correct. After training, 65% of the kept feedback was classified correctly by CLEAR, suggesting that CLEAR can improve human data through filtering, even when the algorithm is trained on incorrect simulated data.

The algorithm did discard some feedback, determining that it was of unknown quality. There were a total of 120 instances of feedback collected on Amazon Mechanical Turk. For each one out of ten participants, we tested over each run (100 total) and all states visited in the human study (12 total). Thus we overall examined 12000 feedback instances. For these states, over 100 runs, 3420 feedback instances were discarded, while 8580 were kept. CLEAR learned more quickly than Policy Shaping and Q-Learning, as shown in Figure 6, which shows the pretraining in simulation to achieve fully trained classifiers before testing the human data. Furthermore, CLEAR keeps learning after PS and Q-Learning have settled on a sub-optimal policy.

VI. Conclusions and Future Work

In this work, we propose Classification for Learning Erroneous Assessments using Rewards (CLEAR), a feedback filter for HRL algorithms. Our results show that when HRL algorithms do not have prior knowledge of the correctness of a feedback source, using CLEAR to estimate better quality feedback improves performance. CLEAR performs similarly over different levels of feedback quality, while Policy Shaping is quite sensitive to the feedback quality. Applying this feedback filtering algorithm can improve expected performance when the robot does not know the quality of feedback ahead of time, which is likely in the wild.

Future work in this area should test how CLEAR performs with real human teachers without the addition of simulated feedback. On large tasks, CLEAR should be tested to determine how much human feedback is needed. Even inexpert human feedback can be expensive to obtain, so methods that require smaller amounts of data can be more effective in the real world. CLEAR could also be tested in much larger, continuous state spaces, using more complex methods than Q-Learning as a baseline. In this case, the classifier C_{CLEAR} could be replaced with a neural net classifier if needed. Finally, testing CLEAR on a non-simulated robot with noise in the learning process would further cement the applicability of this work to the real world.

REFERENCES

[1] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 4, pp. 337–349, 2019.

[2] 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.

[3] M. E. Taylor, N. Nissen, Y. Wang, and N. Navidi, "Improving reinforcement learning with human assistance: an argument for human subject studies with hippo gym," *Neural Computing and Applications*, pp. 1–11, 2021.

[4] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[5] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[6] G. Warnell, N. Waytowich, V. Lawhern, and P. Stone, "Deep tamer: Interactive agent shaping in high-dimensional state spaces," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] W. B. Knox and P. Stone, "Reinforcement learning from simultaneous human and mdp reward," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 475–482.

[8] K. Subramanian, C. L. Isbell Jr, and A. L. Thomaz, "Exploration from demonstration for interactive reinforcement learning," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 447–456.

[9] E. Senft, S. Lemaignan, P. E. Baxter, T. Belpaeme, *et al.*, "Sparc: an efficient way to combine reinforcement learning and supervised autonomy," in *Future of Interactive Learning Machines Workshop at NIPS'16*, 12 2016.

[10] S. Krening and K. M. Feigh, "Newtonian action advice: Integrating human verbal instruction with reinforcement learning," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 720–727.

[11] ——, "Interaction algorithm effect on human experience with reinforcement learning," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 7, no. 2, p. 16, 2018.

[12] Z. Lin, B. Harrison, A. Keech, and M. O. Riedl, "Explore, exploit or listen: Combining human feedback and policy model to speed up deep reinforcement learning in 3d worlds," *arXiv e-prints*, pp. arXiv–1709, 2017.

[13] W. Saunders, G. Sastry, A. Stuhlmueller, and O. Evans, "Trial without error: Towards safe reinforcement learning via human intervention," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 2067–2069.

[14] G. Li, B. He, R. Gomez, and K. Nakamura, "Interactive reinforcement learning from demonstration and human evaluative feedback," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2018, pp. 1156–1162.

[15] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep q-learning from demonstrations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6292–6299.

[17] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv e-prints*, pp. arXiv–1709, 2017.

[18] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, 2017, pp. 4299–4307.

[19] A. Kurenkov, A. Mandlekar, R. Martin-Martin, S. Savarese, and A. Garg, "Ac-teach: A bayesian actor-critic method for policy learning with an ensemble of suboptimal teachers," in *Conference on Robot Learning (CoRL)*, 2019.

[20] M. Sridharan, "Augmented reinforcement learning for interaction with non-expert humans in agent domains," in *2011 10th International Conference on Machine Learning and Applications and Workshops*, vol. 1. IEEE, 2011, pp. 424–429.

[21] T. Kessler Faulkner, E. S. Short, and A. L. Thomaz, "Interactive reinforcement learning with inaccurate feedback," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Submitted for review.

[22] D. Brown, W. Goo, P. Nagarajan, and S. Niekum, "Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations," in *International conference on machine learning*. PMLR, 2019, pp. 783–792.

[23] D. H. Grollman and A. G. Billard, "Robot learning from failed demonstrations," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 331–342, 2012.

[24] J. Zheng, S. Liu, and L. M. Ni, "Robust bayesian inverse reinforcement learning with sparse behavior noise," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

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

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

[27] T. Cederborg, I. Grover, C. L. Isbell, and A. L. Thomaz, "Policy shaping with human teachers," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.