Rating-based Reinforcement Learning

Devin White [1] **Mingkang Wu** [1] **Ellen Novoseller** [2] **Vernon Lawhern** [2] **Nick Waytowich** [2] **Yongcan Cao** [1]

Abstract

This paper develops a novel rating-based reinforcement learning approach that uses human ratings to obtain human guidance in reinforcement learning. Different from the existing preference-based and ranking-based reinforcement learning paradigms, based on human relative preferences over sample pairs, the proposed rating-based reinforcement learning approach is based on human evaluation of individual trajectories without relative comparisons between sample pairs. The rating-based reinforcement learning approach builds on a new prediction model for human ratings and a novel multi-class loss function. We conduct several experimental studies based on synthetic ratings and real human ratings to evaluate the effectiveness and benefits of the new rating-based reinforcement learning approach.

1. Introduction

With the development of deep neural network theory and improvements in computing hardware, deep reinforcement learning (RL) has become capable of handling complex tasks with large state and/or action spaces (*e.g.*, Go and Atari games) and yielding human-level or better-than-human-level performance (Silver et al., 2016; Mnih et al., 2015). Numerous approaches, such as DQN (Mnih et al., 2015), DDPG (Lillicrap et al., 2015), PPO (Schulman et al., 2017), and SAC (Haarnoja et al., 2018) have been developed to address challenges such as stability, exploration, and convergence for various applications (Li, 2019) such as robotic control, autonomous driving, and gaming. Despite the important and fundamental advances behind these algorithms, one key obstacle for the wide application of deep RL is the required knowledge of a reward function, which is often unavailable in practical applications.

Although human experts could design reward functions in some domains, the cost is high because human experts need to understand the relationship between the mission objective and state-action values and may need to spend extensive time adjusting reward parameters and trade-offs not to encounter adverse behaviors such as reward hacking (Amodei et al., 2016). Another approach is to utilize qualitative human inputs *indirectly* to learn a reward function, such that humans guide reward function design *without* directly handcrafting the reward. Existing work on reward learning includes inverse reinforcement learning (IRL) (Ziebart et al., 2008), preference-based reinforcement learning (PbRL) (Christiano et al., 2017), and the combination of demonstrations and relative preferences, e.g. learning from preferences over demonstrations (Brown et al., 2019).

IRL seeks to infer reward functions from demonstrations such that the learned reward functions generate behaviors that are similar to the demonstrations. Numerous IRL methods (Ng et al., 2000), such as maximum entropy IRL (Ziebart et al., 2008; Wulfmeier et al., 2015), nonlinear IRL (Finn et al., 2016), Bayesian IRL methods (Levine et al., 2011; Choi & Kim, 2011; 2012), adversarial IRL (Fu et al., 2018), and behavioral cloning IRL (Szot et al., 2022) have been developed to infer reward functions. The need for demonstrations often makes these IRL methods costly since human experts are needed to provide demonstrations.

Instead of requiring human demonstrations, PbRL (Wirth et al., 2017; Christiano et al., 2017; Ibarz et al., 2018; Liang et al., 2022; Zhan et al., 2021; Xu et al., 2020; Lee et al., 2021b; Park et al., 2022) leverages human pairwise preferences over trajectory pairs to learn reward functions. Querying humans for pairwise preferences rather than demonstrations can dramatically save human time. In addition, by leveraging techniques such as adversarial neural networks (Zhan et al., 2021), additional human time can be saved by learning a well-performing model to predict human preference. Another benefit of PbRL is that humans can provide preferences with respect to uncertainty to promote exploration (Liang et al., 2022). Despite these benefits, PbRL can be ineffective, especially for complex environments, because pairwise preferences only provide relative information rather than directly evaluating sample quality; while in some domains, sampled pairs may be selected carefully to infer global information, in practice, even if one

[1]Department of Electrical and Computer Engineering, University of Texas, San Antonio, TX, United States. [2]Army Research Lab, Adelphi, MD, United States. Correspondence to: Nick Waytowich <nicholas.r.waytowich.civ@army.mil>.

The Many Facets of Preference Learning Workshop at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

sample is preferred over another, it does not necessarily mean that this sample is good. People can also have difficulty when comparing similar samples, thus taking more time and potentially yielding inaccurate preference labels. Notably, several works have sought to improve sample efficiency of PbRL; for instance, PEBBLE (Lee et al., 2021b) considers off-policy PbRL and SURF (Park et al., 2022) explores data augmentations in PbRL.

Other methods for learning reward functions from humans include inferring rewards via rankings over a pool of demonstrations (Brown et al., 2020a; 2019; 2020b) to extrapolate better-than-demonstrator performance from the learned rewards. Other methods of combining pairwise preferences and demonstrations include first learning from demonstrations and then fine-tuning with preferences (Ibarz et al., 2018; Bıyık et al., 2022a). Finally, in the TAMER framework (Knox & Stone, 2009; Warnell et al., 2018; Celemin & Ruiz-del Solar, 2015), a person gives positive (encouraging) and negative (discouraging) feedback to an agent with respect to specific states and actions, instead of over entire trajectories. These methods generally take actions greedily with respect to the learned reward, which may not yield an optimal policy in continuous control settings.

Existing human-guided reward learning approaches have demonstrated effective performance in various tasks. However, they suffer from some key limitations. For example, IRL requires expert demonstrations and hence, cannot be directly applied to tasks that are difficult for humans to demonstrate. PbRL is a practical approach to learning rewards for RL, since it is straightforward for humans to provide accurate relative preference information. Yet, RL from pairwise preferences suffers from some key disadvantages. First, each pairwise preference provides only a single bit of information, which can result in sample inefficiency. In addition, due to their binary nature, standard preference queries do not indicate how much better or worse one sample is than another. Furthermore, because preference queries are relative, they cannot directly provide a global view of each sample's absolute quality (good vs. bad); for instance, if all choices shown to the user are of poor quality, the user cannot say, "A is better than B, but they're both bad!". Thus, a PbRL algorithm may be more easily trapped in a local optimum, and cannot know to what extent its performance approaches the user's goal. Finally, PbRL methods often require strict preferences, such that comparisons between similar-quality or incomparable trajectories cannot be used in reward learning. While some works use weak preference queries (Bıyık et al., 2020; 2022b), in which the user can state that two choices are equally preferable, there is no way to specify the quality (good vs. poor) of such trajectories; thus, valuable information remains untapped.

The objective of the paper is to design a new rating-based RL (RbRL) approach that infers reward functions via multi-class human ratings. RbRL differs from IRL and PbRL in that it leverages human ratings on individual samples, whereas IRL uses demonstrations and PbRL uses relative pairwise comparisons. In each query, RbRL displays one trajectory to a human and requests the human to provide a discrete rating. The number of rating classes can be as low as two, e.g. "bad" and "good", and can be as high as desired. It is worth mentioning that ratings, a well-received concept by humans, can provide informative absolute evaluations of samples. In contrast, the pairwise comparisons used in PbRL can only provide relative comparisons between sample pairs. Hence, ratings provide rich evaluative information for reward function learning.

The main contributions of this paper are as follows. First, we propose a novel RbRL framework for reward function and policy learning from qualitative, absolute human evaluations. Second, we design a new multi-class cross-entropy loss function that accepts multi-class human ratings as the input. The new loss function is based on the computation of a relative episodic reward index and the design of a new multi-class probability distribution function based on this index. Third, we conduct several experiment studies to quantify the impact of the number of rating classes on the performance of RbRL, and compare RbRL and PbRL under both synthetic and real human feedback. Our studies suggest that (1) too few or many rating classes can be disadvantageous, (2) RbRL can outperform PbRL under both synthetic and real human feedback, and (3) people find RbRL to be less demanding, discouraging, and frustrating than PbRL.

2. Problem Formulation

We consider a Markov Decision process without reward (MDP $\setminus$ R) augmented with ratings, which is a tuple of the form $(\mathcal{S}, \mathcal{A}, T, \rho, \gamma, n)$. Here, $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of possible actions, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a state transition probability function specifying the probability $p(s' \mid s, a)$ of reaching state $s' \in \mathcal{S}$ after taking action a in state s, $\rho : \mathcal{S} \to [0, 1]$ specifies the initial state distribution, γ is a discount factor, and n is the number of rating classes. The learning agent interacts with the environment through rollout trajectories, where a length-k trajectory segment takes the form $(s_1, a_1, s_2, a_2, \ldots, s_k, a_k)$. A *policy* π is a function that maps states to actions, such that $\pi(a \mid s)$ is the probability of taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$.

In traditional RL, the environment would receive a reward signal $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, mapping state-action pairs to a numerical reward, such that at time-step t, the algorithm receives a reward $r_t = r(s_t, a_t)$, where (s_t, a_t) is the state-action pair at time t. Accordingly, the standard RL problem can be formulated as a search for the optimal policy π^*, where $\pi^* = \arg\max_\pi \sum_{t=0}^{\infty} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \left[\gamma^t r(s_t, a_t)\right]$,

$a_t \sim \pi(\cdot|s_t)$, and ρ_π is a marginal state-action distribution given a policy π. Note that standard RL assumes the availability of the reward function r. When such a reward function is unavailable, standard RL and its variants may not be used to derive control policies. Instead, we assume that the user can assign any given trajectory segment $\tau = (s_1, a_1, \ldots, s_k, a_k)$ a rating in the set $\{0, 1, \ldots, n-1\}$ indicating the quality of that segment, where 0 is the lowest possible rating, while $n-1$ is the highest possible rating.

The algorithm presents a series of trajectory segments σ to the human and receives corresponding human ratings. Let $X := \{(\sigma_i, c_i)\}_{i=1}^{l}$ be the dataset of observed human ratings, where $c_i \in \{0, \ldots, n-1\}$ is the rating class assigned to segment σ_i, and l is the number of rated segments contained in X at the given point during learning.

Note that descriptive labels can also be given to the rating classes. For example, for $n = 4$ rating classes, we can call the rating class 0 "very bad", the rating class 1 "bad", the rating class 2 "good", and the rating class 3 "very good". With $n = 3$ rating classes, we can call the rating class 0 " bad", the rating class 1 "neutral", and class 2 "good".

3. Rating-based Reinforcement Learning

3.1. Modeling Reward and Return

Our approach learns a reward model $\hat{r} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that predicts state-action rewards $\hat{r}(s, a)$. We further define $\hat{R}(\sigma) := \sum_{t=1}^{k} \gamma^t \hat{r}(s_t, a_t)$ as the cumulative discounted reward, or the *return*, of length-k trajectory segment σ. Larger $\hat{R}(\sigma)$ corresponds to a higher predicted human rating for segment σ. Next, we define $\tilde{R}(\sigma)$ as a function mapping a trajectory segment σ to an estimated total discounted reward, normalized to fall in the interval $[0, 1]$ based on the dataset of rated trajectory segments X:

$$\tilde{R}(\sigma) = \frac{\hat{R}(\sigma) - \min_{\sigma' \in X} \hat{R}(\sigma')}{\max_{\sigma' \in X} \hat{R}(\sigma') - \min_{\sigma' \in X} \hat{R}(\sigma')}. \tag{1}$$

3.2. Novel Rating-Based Cross-Entropy Loss Function

To construct a new (cross-entropy) loss function that can take multi-class human ratings as the input, we need to estimate the human's rating class predictions. We here propose a new multi-class cross-entropy loss given by:

$$L(\hat{r}) = -\sum_{\sigma \in X} \left(\sum_{i=0}^{n-1} \mu_\sigma(i) \log \left(Q_\sigma(i) \right) \right), \tag{2}$$

where X is the collected dataset of user-labeled segments, $\mu_\sigma(i)$ is an indicator that equals 1 when the user assigns rating i to trajectory segment σ, and $Q_\sigma(i) \in [0, 1]$ is the estimated probability that the human assigns the segment σ to the ith rating class. Next, we will model the probabilities $Q_\sigma(i)$ of the human choosing each rating class. Notably, we do this *without* comparing the segment σ to other segments.

3.3. Modeling Human Rating Probabilities

We next describe our model for $Q_\sigma(i)$ based on the normalized predicted returns $\tilde{R}(\sigma)$. To model the probability that σ belongs to a particular class, we will first model separations between the rating classes in reward space.

We define rating class boundaries $\bar{R}_0, \bar{R}_1, \ldots, \bar{R}_n$ in the space of normalized trajectory returns such that $0 := \bar{R}_0 \leq \bar{R}_1 \leq \ldots \leq \bar{R}_n := 1$. Then, if a segment σ has normalized predicted return $\tilde{R}(\sigma)$ such that $\bar{R}_i \leq \tilde{R}(\sigma) \leq \bar{R}_{i+1}$, we wish to model that σ belongs to rating class i with the highest probability.

For example, when the total number of rating classes is $n = 4$, we aim to model the lower and upper return bounds for rating classes $0, 1, 2,$ and 3, which could respectively correspond to "very bad", "bad", "good", and "very good". In this case, if $0 \leq \tilde{R}(\sigma) < \bar{R}_1$, then we would like our model to predict that σ most likely belongs to class 0 ("very bad"), while if $\bar{R}_2 \leq \tilde{R}(\sigma) < \bar{R}_3$, then our model should predict that σ most likely belongs to class 2 ("good").

We propose to model $Q_\sigma(i)$ given the normalized predicted returns $\tilde{R}(\sigma)$ and rating category separations $\bar{R}_i$:

$$Q_\sigma(i) = \frac{e^{-k(\tilde{R}(\sigma) - \bar{R}_i)(\tilde{R}(\sigma) - \bar{R}_{i+1})}}{\sum_{j=0}^{n-1} e^{-k(\tilde{R}(\sigma) - \bar{R}_j)(\tilde{R}(\sigma) - \bar{R}_{j+1})}}, \tag{3}$$

where k is a hyperparameter modeling the noisiness in the human labels, and the denominator ensures that $\sum_{i=0}^{n-1} Q_\sigma(i) = 1$, i.e. that the class probabilities sum to 1.

To gain intuition for Equation (3), note that when $\tilde{R}(\sigma) \in (\bar{R}_i, \bar{R}_{i+1})$, such that the predicted return falls within rating class i's predicted boundaries, then $-(\tilde{R}(\sigma) - \bar{R}_i)(\tilde{R}(\sigma) - \bar{R}_{i+1}) \geq 0$ while $-(\tilde{R}(\sigma) - \bar{R}_j)(\tilde{R}(\sigma) - \bar{R}_{j+1}) \leq 0$ for all $j \neq i$. This means that $Q_\sigma(i) \geq Q_\sigma(j),\ j \neq i$, so that the model assigns category i the highest class probability, as desired. Furthermore, we note that $Q_\sigma(i)$ is maximized when $\tilde{R}(\sigma) = \frac{1}{2}(\bar{R}_i + \bar{R}_{i+1})$, such that the predicted return falls directly in the center of category i's predicted range. As $\tilde{R}(\sigma)$ becomes increasingly further from $\frac{1}{2}(\bar{R}_i + \bar{R}_{i+1})$, the modeled probability $Q_\sigma(i)$ of class i monotonically decreases. These probability trends are illustrated in Figure 7 in the Appendix B. We next show how to compute the class boundaries $\bar{R}_i,\ i = 0, \ldots, n$.

Modeling Boundaries between Rating Categories. Next, we discuss how to model the boundaries between rating categories, $0 =: \bar{R}_0 \leq \bar{R}_1 \leq \ldots \leq \bar{R}_n := 1$. We determine these boundary values based on the distribution of $\tilde{R}(\sigma)$ for the trajectory segments $\sigma \in X$ and the number of observed samples in X from each rating class. We select the $\bar{R}_i$ values such that the number of training data samples that the model assigns to each modeled rating class matches the number of samples in X that the human assigned to that rating class. Note that this does not require the predicted ratings based on $\tilde{R}(\sigma)$ to match the human ratings for σ in the training data X, but ensures that the proportions of segments assigned to each rating class matches that in the human dataset X. This matching in rating class proportions is desirable for learning an appropriate reward function based on human preference, since different humans could give ratings in significantly different proportions depending on their preferences and latent reward functions, as modeled by $\hat{R}$.

To define each $\bar{R}_i$ so that the number of samples in each modeled rating category reflects the numbers of ratings in the human data, we first sort the estimated returns $\tilde{R}(\sigma)$ for all $\sigma \in X$ from lowest to highest, and label these sorted estimates as $\tilde{R}_1 \leq \tilde{R}_2 \leq \cdots \leq \tilde{R}_l$, where l is the cardinality of X. Denoting via k_j the number of segments that the human assigned to rating class j, $j \in \{0, \cdots, n-1\}$, we can then model each category boundary $\bar{R}_i, i \notin \{0, n\}$ (since $\bar{R}_0 = 0$ and $\bar{R}_n = 1$ by definition), as follows:

$$\bar{R}_i = \frac{\tilde{R}_{k_i^{\text{cum}}} + \tilde{R}_{1+k_i^{\text{cum}}}}{2}, \quad i \in \{1, 2, \ldots, n-1\}, \qquad (4)$$

where $k_i^{\text{cum}} := \sum_{j=0}^{i} k_j$ is the total number of segments that the human assigned to any rating category $j \leq i$. When the user has not assigned any ratings within a particular category, i.e., $k_i = 0$ for some i, then we define the upper bound for category i as $\bar{R}_{k_{i+1}} := \bar{R}_{k_i}$, such the lower and upper bounds for this rating category are identical.

This definition guarantees that when all normalized return predictions $\tilde{R}(\sigma), \sigma \in X$, are distinct, then our model places k_i segments within each interval $[\bar{R}_i, \bar{R}_{i+1})$ for $i < n-1$ and k_i segments in $[\bar{R}_i, \bar{R}_{i+1}]$ for $i = n-1$, and thus predicts that k_i segments have rating i.

4. Synthetic Experiments

4.1. Setup

We conduct synthetic experiments based on the setup in Lee et al. (2021a) to evaluate RbRL relative to the PbRL baseline (Lee et al., 2021a). The goal is to learn to perform a task by obtaining feedback from a teacher, in this case a synthetic human. For the PbRL baseline, we generate synthetic feedback such that in each queried pair of segments, the segment with the higher ground truth cumulative reward is preferred. In contrast to the synthetic preferences between sample pairs in Lee et al. (2021a), RbRL was given synthetic ratings generated for individual samples, where these ratings were given by comparing the sample's ground truth return to ground truth rating class boundaries. For simplicity, we selected these ground truth rating class boundaries so that rating classes are evenly spaced in reward space.

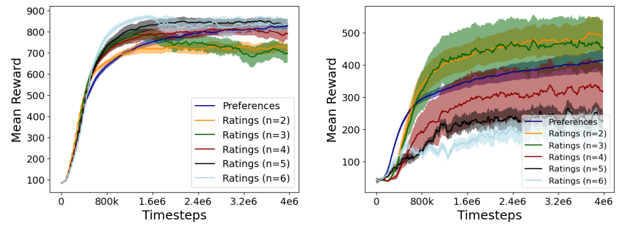


Figure 1. Performance of RbRL in synthetic experiments for different n, compared to PbRL: mean reward $\pm$ standard error over 10 experiment runs for Walker (left) and Quadruped (right).

For the synthetic PbRL experiments, we selected preference queries using the ensemble disagreement approach in Lee et al. (2021a). We extend this method to select rating queries for the synthetic RbRL experiments. First, we train a reward predictor ensemble and obtain the predicted reward for every candidate segment and ensemble member. We then select the segment with the largest standard deviation over the ensemble to receive a rating label. 1000 and 2000 queries were provided for Walker and Quadruped, respectively.

We used the same neural network structures for both the reward predictor and control policy and the same hyperparameters as Lee et al. (2021a).

4.2. Results

Figure 1 shows the performance of RbRL for different number of rating classes (*i.e.* values of n) and PbRL for two environments from Lee et al. (2021a): Walker and Quadruped. We observe that a higher number of rating classes yields better performance for Walker. In addition, RbRL with $n = 5, 6$ outperforms PbRL. However, for Quadruped, while RbRL with $n = 2, 3$ still outperforms PbRL, a higher number of rating classes decreases performance; this decrease may be caused by the selection of rating class boundaries used to generate the synthetic feedback. The results indicate that RbRL is effective and can provide better performance than PbRL even if synthetic ratings feedback is generated using reward thresholds that are evenly distributed, without further optimization of their selection. We expect further optimization of the boundaries used to generate synthetic feedback to yield improved performance.

5. Human Experiments

5.1. Setup

We conduct all human experiments by following a similar setup to Christiano et al. (2017). In particular, the goal is to learn to perform a given task by obtaining feedback from a teacher, in this case a human. Different from PbRL in Christiano et al. (2017), which asks humans to provide their preferences between sample pairs, typically in the form of short video segments, RbRL asks humans to evaluate individual samples, also in the form of short video segments, to provide their ratings, e.g., "segment performance is good" or "segment performance is bad".

For all human experiments, the same neural network structures for both the reward predictor and control policy and the same hyperparameters as in Christiano et al. (2017).

5.2. RbRL with Different Numbers of Rating Classes

To evaluate the impact of the number of rating classes n on RbRL's performance, we conduct tests in which a human expert (an author on the study) provides ratings with $n = 2, \ldots, 8$, in the Cheetah MuJoCo environment. In particular, three experiment runs were conducted for each $n \in \{2, 3, \ldots, 8\}$. Fig. 2 shows the performance of RbRL for each n. It can be observed that RbRL performs better for $n \in \{3, 4, \ldots, 7\}$ than for $n = 2, 8$, indicating that allowing more rating classes is typically beneficial. However, an overly-large n may yield worse performance due to the difficulty of distinguishing between neighboring classes. Additionally, when $n <= 8$, it is possible to assign physical meanings to each n. Table 1 in Appendix C shows the physical meaning of rating classes for $n = 2, 3, \ldots, 8$ used in these experiments. For $n > 8$, it becomes difficult to assign such physical meaning and hence it will be more challenging for users to provide consistent ratings.

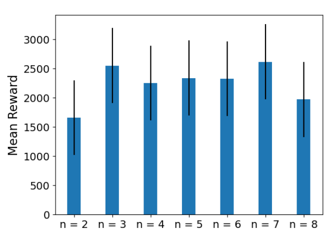


Figure 2. RbRL performance for different n in a human experiment: performance in the Cheetah environment (mean $\pm$ standard error over 3 experiment runs).

5.3. RbRL Human User Study

To evaluate the effectiveness of RbRL for non-expert users, we conducted an IRB-approved human user study. We conducted tests on 3 of the OpenAI Gym MuJoCo Environments also used in Christiano et al. (2017): Swimmer, Hopper and Cheetah. A total of 20 participants were recruited (7 for Cheetah, 7 for Swimmer, and 6 for Hopper). For Cheetah, the goal is to move the agent to the right as fast as possible; this is the same goal encoded in the default hand-crafted environment reward. Similarly, the goal for Swimmer matches that of the default hand-crafted environment reward. However, for Hopper, the goal is to teach the agent to do a backflip, which differs from the goal encoded by the default hand-crafted environment reward. We chose to study the back flip task to see how well RbRL can learn new behaviors for which a reward is unknown.

During the user study, each participant performed two tests—one for RbRL and one for PbRL—in one of the three MuJoCo environments, both for $n = 2$ rating classes. To eliminate potential bias, we assigned each participant a randomized order in which to perform the PbRL and RbRL experiment runs. Because the participants had no prior knowledge of the MuJoCo environments tested, we provided sample videos to show desired and undesired behaviors so that the participants could better understand the task. For each experiment run, the participant was given 30 minutes to given rating/preference labels. Once finished, the participant filled out a questionnaire about the tested algorithm. The participant was then given a 10 minute break before conducting another test and complete the questionnaire about the other algorithm. Afterwards, the participant completed the questionnaire about comparison between the two algorithms. The questionnaire can be found in Appendix D. Policy and reward learning occurred during the 30 minutes in which the user answered queries, and then continued after the human stepped away until the code reached 4 million time-steps.

5.3.1. Performance

Figure 3 shows the performance of PbRL and RbRL across the seven participants for the Cheetah and Swimmer tasks. It can be seen that RbRL performs similarly to or better than PbRL. In particular, RbRL can learn quickly in both cases, evidenced by the fast reward growth early during learning. Figure 3 also displays the results when experts (authors on the study) provided ratings and preferences for Cheetah and Swimmer. For consistency, the same expert tested PbRL and RbRL in each environment. We observe that for the expert trials, RbRL performs consistently better than PbRL in the same human time. This suggests that RbRL can outperform PbRL regardless of the user's domain knowledge of the environments. It can also be observed that the RbRL and PbRL trials with expert users outperform the trials in which feedback is given by non-experts.

Although RbRL performs similarly to PbRL in the Cheetah task, we observed that some participants performed very poorly in this environment, perhaps due to lack of understanding of the task. In Appendix A, the raw data of all participants for Cheetah and Swimmer is provided to show performance under each individual participant's guidance. From the individual results for Cheetah (RbRL), we can see that one of the trials performs very poorly (with the final reward less than -10). For all other tests, including both

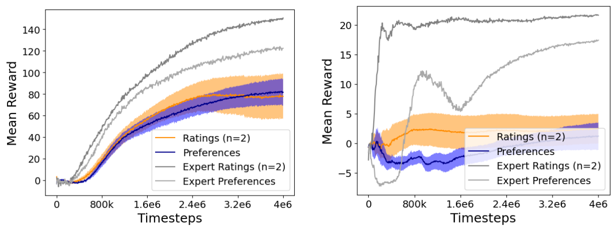


Figure 3. Performance of RbRL in the human user study: Cheetah (left) and Swimmer (right). For non-expert users, the plots show mean $\pm$ standard error over 7 users. The expert results are each over a single experiment run.

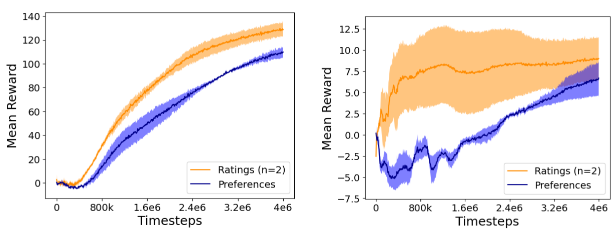


Figure 4. RbRL performance for the top 3 (non-expert) user study participants: mean reward $\pm$ standard error over the 3 experiment runs each for Cheetah (left) and Swimmer (right)

PbRL and RbRL, the final reward is in positive territory, usually more than 20. Figure 4 shows the mean reward for the top 3 non-expert users at different iterations for Cheetah and Swimmer. It can be observed that RbRL consistently outperforms PbRL and learns the goal faster than PbRL.

To compare PbRL and RbRL in the Hopper backflip task, we ran the learned policies for the 6 participants to generate videos. Videos for the best learned policies from PbRL and RbRL can be found at `https://bit.ly/3wgF5OH`, and indicate that (1) both RbRL and PbRL can learn the backflip, and (2) the backflip learned via RbRL fits better with our understanding of a desired backflip.

5.3.2. User Questionnaire Results

To understand how the non-expert users view their experience of giving rating and preferences, we conduct a post-experiment user questionnaire, shown in Appendix D. The questionnaire asked users for feedback about their experience supervising PbRL and RbRL, and their comparison between PbRL and RbRL. Figure 5 displays the normalized survey results from the 20 user study participants. In particular, the left subfigure of Figure 5 shows the participants' responses with respect to their separate opinions about PbRL and RbRL. These responses suggest that PbRL is more demanding and difficult than PbRL, leading users to feel more insecure and discouraged than when using RbRL. The right subfigure of Figure 5 shows the survey responses when users were asked to compare PbRL and RbRL; these results confirm the above findings and also show that users perceive themselves as completing the task more successfully when providing ratings (RbRL). One interesting observation is that the participants prefer RbRL and PbRL equally, which differs from the other findings. However, one participant stated that he/she preferred PbRL more because PbRL is more challenging. This suggests that "like" is a very subjective concept, making the responses for this question less informative than those for the other survey responses.

We also conducted a quantitative analysis of the human time effectiveness when humans were asked to give ratings and preferences. These results are given in Appendix E.

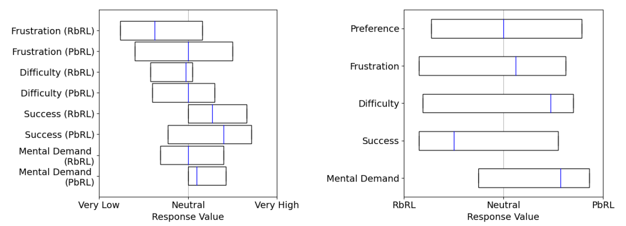


Figure 5. Participants' responses to questions for RbRL and PbRL. The set of survey questions is detailed in the Appendix. The blue bar indicates the median and the edges depict the 1st quartile (left) and 3rd quartile (right).

6. Discussion and Open Challenges

One key difference between PbRL and RbRL is the value of the acquired human data. Because ratings in RbRL are not relative, they have the potential to provide more global value than preferences, especially when queries are not carefully selected. For environments with large state-action spaces, ratings can provide more value for reward learning. One limitation of ratings feedback is that the data samples in different rating classes can be very different, leading to imbalanced datasets. We expect that addressing the data imbalance issue would further improve RbRL performance.

One challenge for RbRL is that ratings may not be given consistently during learning, especially considering users' attention span and fatigue level over time. Future work includes developing mechanisms to quantify users' consistency levels, the impact of user inconsistency, or solutions to user inconsistency.

Another potential limitation of RbRL is that it learns a less refined reward function than PbRL because RbRL does not seek to distinguish between samples from the same rating class. Hence, future work could integrate RbRL and PbRL to create a multi-phase learning strategy, where RbRL provides fast initial global learning while PbRL further refines performance via local queries based on sample pairs.

Acknowledgements

The authors were supported in part by Army Research Lab under grant W911NF2120232, Army Research Office under grant W911NF2110103, and Office of Naval Research under grant N000142212474. We would like to thank Feng Tao, Van Ngo, Gabriella Forbis who provided helpful feedback, code, and tests.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Bıyık, E., Palan, M., Landolfi, N. C., Losey, D. P., Sadigh, D., et al. Asking easy questions: A user-friendly approach to active reward learning. In *Conference on Robot Learning*, pp. 1177–1190, 2020.

Bıyık, E., Losey, D. P., Palan, M., Landolfi, N. C., Shevchuk, G., and Sadigh, D. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research*, 41(1):45–67, 2022a.

Bıyık, E., Talati, A., and Sadigh, D. APReL: A library for active preference-based reward learning algorithms. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 613–617, 2022b.

Brown, D., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *Proceedings of the International Conference on Machine Learning*, pp. 783–792, 2019.

Brown, D., Coleman, R., Srinivasan, R., and Niekum, S. Safe imitation learning via fast Bayesian reward inference from preferences. In *International Conference on Machine Learning*, pp. 1165–1177, 2020a.

Brown, D. S., Goo, W., and Niekum, S. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on Robot Learning*, pp. 330–359, 2020b.

Celemin, C. and Ruiz-del Solar, J. COACH: Learning continuous actions from corrective advice communicated by humans. In *2015 International Conference on Advanced Robotics (ICAR)*, pp. 581–586, 2015.

Choi, J. and Kim, K.-E. Map inference for Bayesian inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 24, 2011.

Choi, J. and Kim, K.-E. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. *Advances in Neural Information Processing Systems*, 25, 2012.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Finn, C., Levine, S., and Abbeel, P. Guided cost learning: Deep inverse optimal control via policy optimization. In *Proceedings of the International Conference on Machine Learning*, pp. 49–58, 2016.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning*, pp. 1861–1870, 2018.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in Atari. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Lee, K., Smith, L., Dragan, A., and Abbeel, P. B-Pref: Benchmarking preference-based reinforcement learning. *Advances in Neural Information Processing Systems*, 2021a.

Lee, K., Smith, L. M., and Abbeel, P. PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*, pp. 6152–6163, 2021b.

Levine, S., Popovic, Z., and Koltun, V. Nonlinear inverse reinforcement learning with Gaussian processes. *Advances in Neural Information Processing Systems*, 24, 2011.

Li, Y. Reinforcement learning applications. *arXiv preprint arXiv:1908.06973*, 2019.

Liang, X., Shu, K., Lee, K., and Abbeel, P. Reward uncertainty for exploration in preference-based reinforcement learning. In *International Conference on Learning Representation*, 2022.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *International Conference on Learning Representations*, 2015.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Ng, A. Y., Russell, S. J., et al. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 2, 2000.

Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., and Lee, K. SURF: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In *International Conference on Learning Representations*, 2022.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

Szot, A., Zhang, A., Batra, D., Kira, Z., and Meier, F. BC-IRL: Learning generalizable reward functions from demonstrations. In *The Eleventh International Conference on Learning Representations*, 2022.

Warnell, G., Waytowich, N., Lawhern, V., and Stone, P. Deep TAMER: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Wirth, C., Akrour, R., Neumann, G., Fürnkranz, J., et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136): 1–46, 2017.

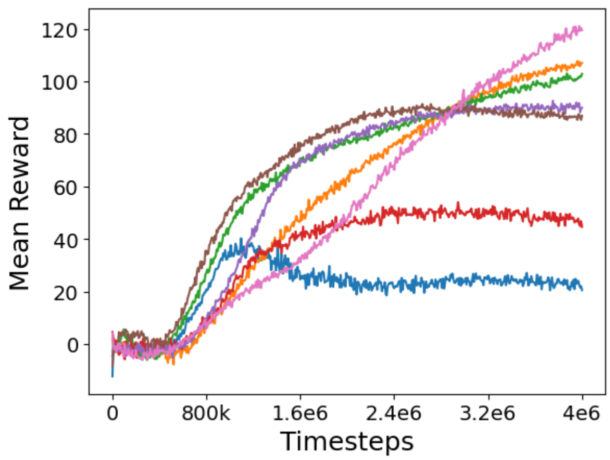
Wulfmeier, M., Ondruska, P., and Posner, I. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

Xu, Y., Wang, R., Yang, L., Singh, A., and Dubrawski, A. Preference-based reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 33:18784–18794, 2020.

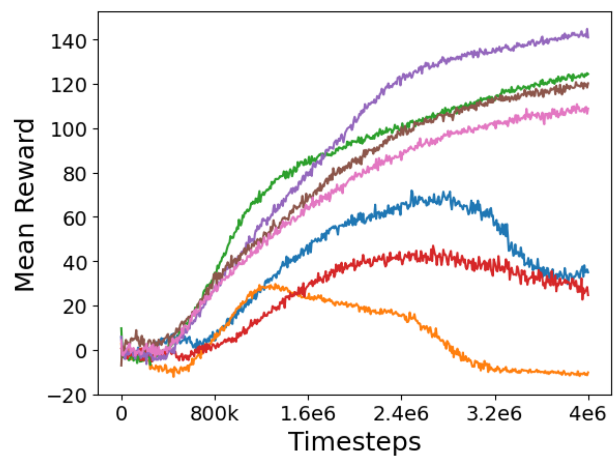
Zhan, H., Tao, F., and Cao, Y. Human-guided robot behavior learning: A GAN-assisted preference-based reinforcement learning approach. *IEEE Robotics and Automation Letters*, 6(2):3545–3552, 2021.

Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2008.

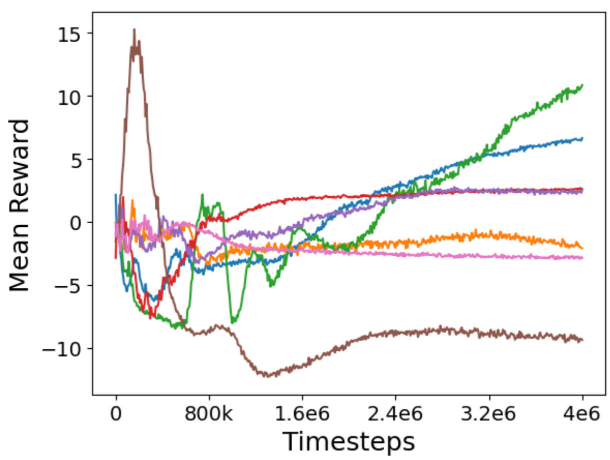
A. Raw Data for Agent Performance from Individual Participants in the User Study



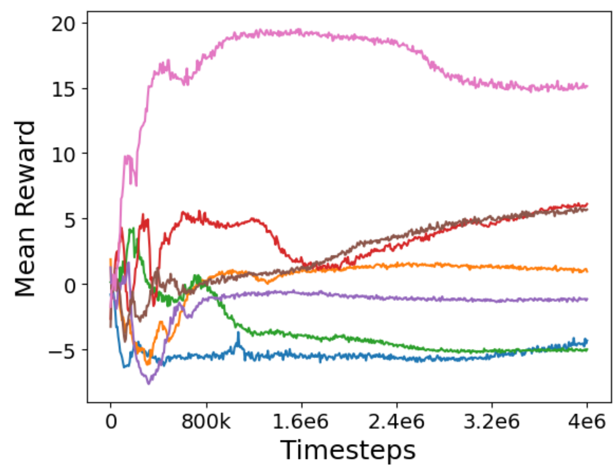
(a) Individual results for Cheetah (PbRL)



(b) Individual results for Cheetah (RbRL with $n = 2$)



(c) Individual results for Swimmer (PbRL)



(d) Individual results for Swimmer (RbRL with $n = 2$)

Figure 6. Results of individual runs in the human user study for the Cheetah and Swimmer environments (7 users in each case).

B. Modeling Rating Class Probabilities

Figure 7 provides an intuitive illustration of the modeled class probabilities $Q_\sigma(i)$, as calculated via Equation (3).

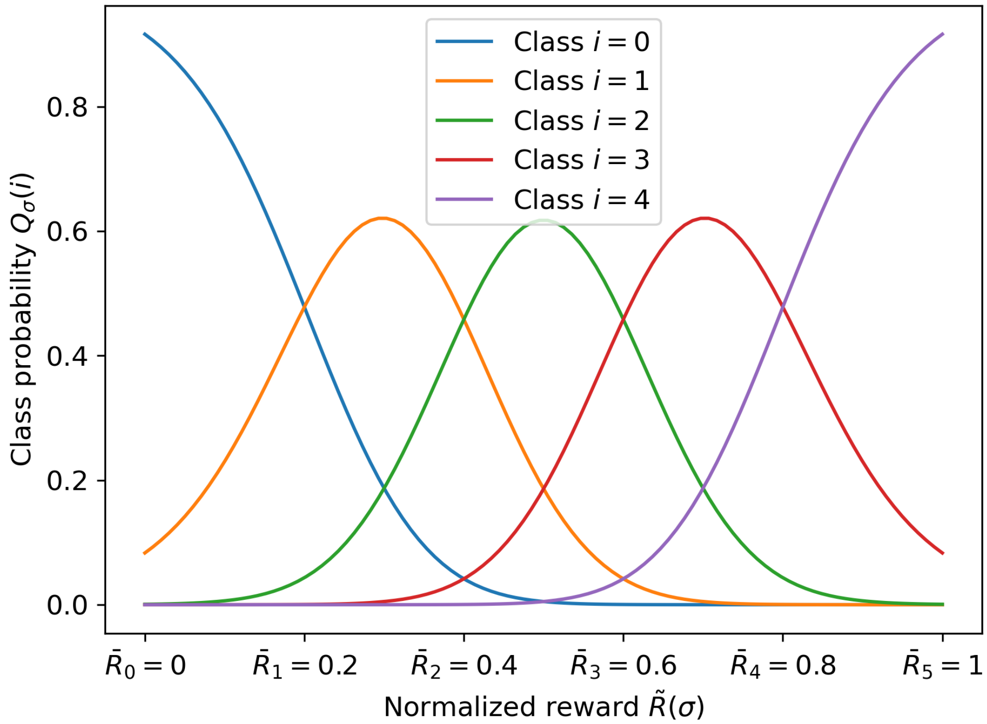


Figure 7. RbRL's modeled class probabilities $Q_\sigma(i)$. This figure illustrates Equation (3), evaluated with $n = 5$ classes, $k = 30$, and the rating category separation parameters hard-coded at $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$. We can see that the probability of belonging to a class i is maximized when the normalized return $\tilde{R}(\sigma)$ is halfway between $\bar{R}_i$ and $\bar{R}_{i+1}$, and decreases as $\tilde{R}(\sigma)$ moves further from this point.

C. Physical Meaning for Different n

Table 1 displays the physical meanings of the rating classes used in our human experiments for all numbers of rating classes (values of n) considered.

Number of Rating Classes	Rating Classes
$n = 2$	"good", "bad"
$n = 3$	"good", "neutral","bad"
$n = 4$	"very good", "good", "bad", "very bad"
$n = 5$	"very good", "good", "neutral","bad", "very bad"
$n = 6$	"very good", "good", "slightly good", "slightly bad","bad", "very bad"
$n = 7$	"very good", "good", "slightly good", "neutral" "slightly bad","bad", "very bad"
$n = 8$	"perfect", "very good", "good", "slightly good", "neutral" "slightly bad","bad", "very bad"

Table 1. Physical meaning of rating classes for $n = 2, 3, \ldots, 8$ used in our human experiments.

D. Questionnaire

The three-part questionnaire given to the user study participants appears next.

User Study Questionnaire

Instructions: Mark the bin best reflecting your response when instructed to do so.

Rating-based feedback:

1. How mentally demanding was the task?

Very low Neutral Very high

2. How successful were you in accomplishing what you were asked to do?

Very low Neutral Very high

3. How hard did you have to work to accomplish your level of performance?

Very low Neutral Very high

4. How insecure, discouraged, irritated, stressed, and annoyed were you?

Very low Neutral Very high

User Study Questionnaire

Instructions: Mark the bin best reflecting your response when instructed to do so.

Preference-based feedback:

1. How mentally demanding was the task?

Very low Neutral Very high

2. How successful were you in accomplishing what you were asked to do?

Very low Neutral Very high

3. How hard did you have to work to accomplish your level of performance?

Very low Neutral Very high

4. How insecure, discouraged, irritated, stressed, and annoyed were you?

Very low Neutral Very high

User Study Questionnaire

Instructions: Mark the bin best reflecting your response when instructed to do so.

Comparison of preference-based and ratings-based feedback:

1. Which was more mentally demanding?

Ratings Neutral Preferences

2. With which were you able to better accomplish the task?

Ratings Neutral Preferences

3. With which was it more difficult to accomplish the task?

Ratings Neutral Preferences

4. Which caused you to be more insecure, discouraged, irritated, stressed, and annoyed?

Ratings Neutral Preferences

5. Which did you like more?

Ratings Neutral Preferences

E. Human Time

Figure 8 shows the average number of human queries provided in 30 minutes for Cheetah, Swimmer, Hopper, and all three environments combined. It can be observed that the participants can provide more ratings than pairwise preferences in all environments, indicating that it is easier and more efficient to provide ratings than to provide pairwise preferences. On average, participants can provide approximately 14.03 ratings per minute, while they provide only 8.7 preferences per minute, which means that providing a preference requires 62% more time than providing a rating. For Cheetah, providing a preference requires 100%+ more time than providing a rating, which is mainly due to the need to compare video pairs that are very similar. For Swimmer and Hopper, the environments and goals are somewhat more complicated. Hence, providing ratings can be slightly more challenging, but is still easier than providing pairwise preferences.

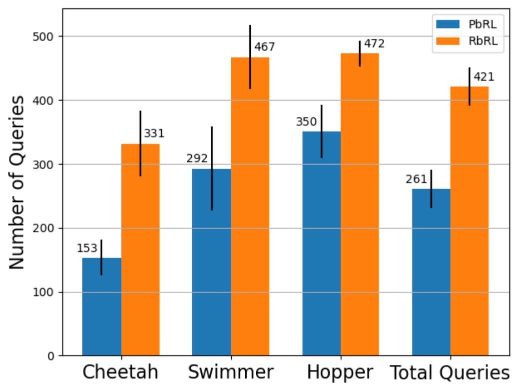


Figure 8. Number of queries provided in 30 minutes across participants in our human user study (mean ± standard error).