

## Data Driven Reward Initialization for Preference based Reinforcement Learning

**Mudit Verma<sup>1</sup> Subbarao Kambhampati<sup>1</sup>**

<sup>1</sup> SCAI, Arizona State University  
muditverma@asu.edu, rao@asu.edu

## Abstract

Preference-based Reinforcement Learning (PbRL) methods utilize binary feedback from the human in the loop (HiL) over queried trajectory pairs to learn a reward model in an attempt to approximate the human’s underlying reward function capturing their preferences. In this work, we investigate the issue of a high degree of variability in the initialized reward models which are sensitive to random seeds of the experiment. This further compounds the issue of degenerate reward functions PbRL methods already suffer from. We propose a data-driven reward initialization method that does not add any additional cost to the human in the loop and negligible cost to the PbRL agent and show that doing so ensures that the predicted rewards of the initialized reward model are uniform in the state space and this reduces the variability in the performance of the method across multiple runs and is shown to improve the overall performance compared to other initialization methods.

## Introduction

Reinforcement Learning (RL) allows AI agents to learn by trial and error (Mnih et al. 2015; Arulkumaran et al. 2017; Verma et al. 2019). While the dynamics of the interaction are governed by the environment in which the agent executes a chosen action, the only task-relevant learning signal comes from the given reward function. Therefore, several of the recent works on RL (Hadfield-Menell et al. 2017; Vamplew et al. 2018) have pointed out the importance of the specified reward function in achieving a given task. Hence, despite recent successes of Deep Reinforcement Learning for complex high dimensional state and action space domains, prior works have found that specifying reward functions for seemingly “easy” tasks could be extremely hard, to begin with, and this becomes even more challenging with potential issues of reward hacking (Vamplew et al. 2018; Krakovna et al. 2020) that may impact Human-AI trust (Zahedi et al. 2021, 2022). In fact, works in explainable AI literature have attempted at explaining environment dynamics to human in the loop (Sreedharan et al. 2020; Verma, Kharkwal, and Kambhampati 2022; Gopalakrishnan, Verma, and Kambhampati 2021a,b), indicating that requesting detailed reward specifications over low-level state space is challenging for HIL (Verma et al. 2021).

Several solutions to the reward learning problem have been proposed in the past like learning from demonstration (LfD) (Schaal 1996) or advice (Guan et al. 2021; Guan, Verma, and Kambhampati 2020), imitation learning (Hussein et al. 2017) and preference-based reinforcement learning (Verma and Metcalf 2022; Lee, Smith, and Abbeel 2021; Park et al. 2022; Christiano et al. 2017). While each paradigm has its own challenges, Preference-based Reinforcement Learning (PbRL) relies on binary feedback of a human in the loop (HiL) on queried trajectory pairs to convey what the user “wants” or “prefers”, in contrast to a HiL performing a demonstration for imitation learning or LfD line works which could be prohibitively expensive, or human may not have expertise on specific action choices, for example, a human may find it easier to provide their preference over two trajectories regarding which one appears more as a humanoid back-flip than to demonstrate it.

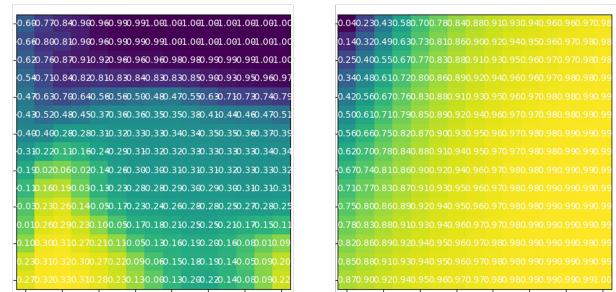


Figure 1: “Patchy” or non-uniform reward prediction by using Kaiming-Uniform weight initialization (left), and the predicted reward values of states by the proposed data-driven reward initialization method over a  $15 \times 15$  gridworld. Numbers in each cell show the maximum reward,  $\bar{R}^h$ , that the agent upon taking an action in that cell.

In Preference-Based Learning methods, recent works (Lee, Smith, and Abbeel 2021; Park et al. 2022; Verma, Kharkwal, and Kambhampati 2022; Christiano et al. 2017; Verma and Metcalf 2022; Soni et al. 2022) have looked at learning the underlying human reward function from human binary feedbacks on trajectory pairs queried by the RL agent. In this work, we focus our attention to the class of PbRL works that attempt to learn (or approximate) a re-

ward model from human preferences. We first discuss that learning a reward model from a finite, possibly small number of trajectory preference queries can potentially lead to degenerate solutions (that is, significantly different, multiple reward models can equally explain the given trajectory preferences, thereby yielding very different agent policies). Now, although this is majorly caused by the limited feedback queries we found that high variability in the learned reward model and subsequently the learned agent policy is also seen when we vary the reward function initialization strategy. We find that the PbRL reward learning process is extremely sensitive to the reward function initialization and changing the random seed for the initialization can produce extremely positive or negative results. In this view, we propose a data-driven reward function initialization technique (inspired from data-driven weight initialization methods (Narkhede, Bartakke, and Sutaone 2022)) that not only stabilizes the resultant performance across multiple runs but also improves the agent’s performance at learning the underlying reward function at potentially zero additional cost to the HiL or the agent being trained.

## Background

Preference-based Reinforcement learning agents sample pairs of trajectories  $\tau_0, \tau_1$  using a random policy or via the policy  $\pi_\phi$  being learned. An image projection of these trajectory pairs is shown to the human (with image space as the established lingua franca (Kambhampati et al. 2022) between the agent and HiL) in the loop where they provide binary feedback to highlight their relative preference of one trajectory over the other, hence populating a buffer  $D_\tau$  with tuples  $\langle \tau_0^i, \tau_1^i, y^i \rangle$ . Extending from a typical reinforcement learning setup, preference-based RL agents solve the reward learning problem,  $P = \langle \mathcal{M} \setminus r, R_\psi^h, \pi^H, Q \rangle$  (Shah et al. 2020) where the agent models the world as a Markov Decision Process  $\mathcal{M} \setminus r = \langle S, T, A \rangle$  without any environment reward, and instead uses a learned  $R_\psi^h$  reward model by querying  $q \in Q$  questions of the form of trajectory pairs to the HiL to obtain binary feedbacks regarding their preference. Recent PbRL works have resorted to supervised learning to solve this problem of reward learning. First, they approximate the underlying human reward model  $\tilde{\mathcal{R}}^h$  by a bounded, parameterized function approximator  $R_\psi^h : S \rightarrow \mathbb{R}$ . They use the Bradley Terry model (Bradley and Terry 1952) to obtain the probability of one trajectory being preferred over another,  $P_\psi$ , as the softmax (Bishop and Nasrabadi 2006) over predicted returns (as the sum of predicted rewards) over the two trajectories.

$$P_\psi[\tau_0 \succ \tau_1] = \frac{\exp(\sum_t R_h(s_t^0, a_t^0))}{\sum_{i \in \{0,1\}} \exp(\sum_t R_h(s_t^i, a_t^i))} \quad (1)$$

This is essentially a classifier to approximate human preference feedbacks which can be trained by minimizing the cross entropy loss between the predictions and the ground truth human labels as follows :

$$\begin{aligned} \mathcal{L}_{CE} = - & \mathbb{E}_{(\tau_0, \tau_1, y) \sim \mathcal{D}} [y(0) \log P_\psi[\tau_0 \\ & \succ \tau_1] + y(1) \log P_\psi[\tau_1 \succ \tau_0]] \end{aligned} \quad (2)$$

Several works have extended the above approach in various ways, by building priors over the reward space, improving the query method, improving the exploration strategy, and state or temporal augmentation to name a few. In this work, we are interested in realizing how susceptible is the backbone algorithm used in several PbRL works to reward initialization and subsequently propose a simple initialization method that poses zero cost to the human in the loop as well as to the PbRL agent.

## Problem

It is well known in the Inverse Reinforcement Learning community that reward model learning suffers from the issue of degeneracy. We discuss reward model degeneracy in the context of preference-based reinforcement learning. Even with a large amount of feedback queries to the human in the loop to elicit their preferences, as pointed out in (Ng, Russell et al. 2000), the base case of  $R_\psi^h(s) = 0 \forall s \in S$  will always be a solution as any policy  $\pi$ , including a random policy as well as the optimal human policy  $\pi_h^*$  according to their underlying reward model  $\tilde{\mathcal{R}}^h$ , is optimal. Without a good inductive bias, we posit that reducing the number of queries to the human in the loop will eventually allow for more degenerate solutions.

However, recent works (Lee, Smith, and Abbeel 2021; Park et al. 2022; Christiano et al. 2017; Verma and Metcalf 2022) all suffer from the issue of degeneracy more than one may expect. In our experiments with (Lee, Smith, and Abbeel 2021) baseline, we found that the method has high variance in the performance of the learned reward model and is very sensitive to extraneous parameters like random seed of the experiment. Now, although several other sources contributing towards the reward function degeneracy exist and may even be unidentified, we found that reward initialization can have a big impact on the subsequent agent’s reward recovery and policy performance.

## Sensitivity to Reward Initialization

Typically, RL (and PbRL) works do not focus on reward function initialization and try to subdue its effects on the final performance by showcasing mean performance over multiple experiments runs. However, in our initial investigation, we found that a change in the initial reward estimates can have a substantial impact on the final performance.

Even with uniform initialization methods (Hu, Xiao, and Pennington 2020; He et al. 2015), we note that the initialization remains uniform in the parameter space  $\psi \sim \Psi$  of the reward model  $R_\psi^h$ , however, the subsequent reward over the domain of the state space under consideration is not at all uniform. In fact, as shown in figure 1 (left), the initial reward values over a 15x15 gridworld space are “patchy” or has certain “peak” state locations and the rewards smoothen out from there. At the very least, such initial rewards heavily bias the sampled trajectories to query the human in the loop whereas in many of the initial queries the given two trajectories are extremely similar contributing to the cognitive load on the human in the loop. In fact, such initialization is assumed to be a “feature than a bug” where works consider

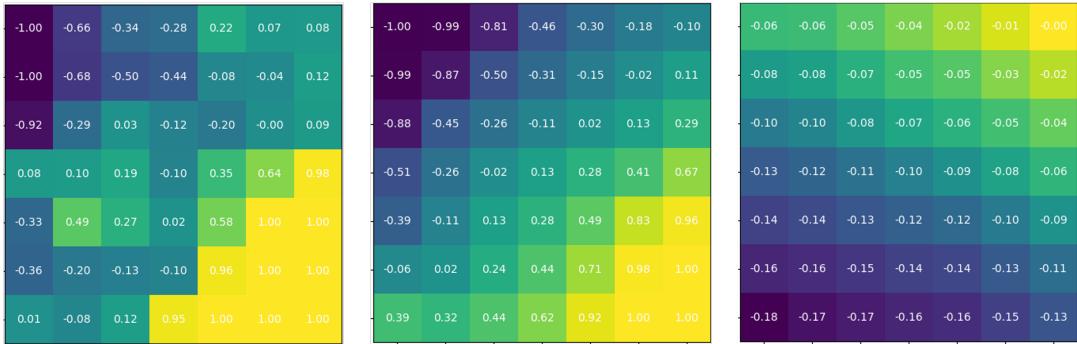


Figure 2: Variations in the initial reward values on 7x7 gridworld domain, when initialized using Xavier-Uniform across three different seeds. Numbers in each cell show the maximum reward,  $\tilde{\mathcal{R}}^h$ , that the agent upon taking an action in that cell.

the initial randomness as a means for the agent to explore in the world. Although we agree that random initial rewards can in fact aid with coming up with diverse trajectories to query to the human in the loop, in reality, it is the reward parameters that are uniform and not the rewards themselves. In fact, we propose that the initialization of reward model parameters such that the corresponding reward values on the domain of the state space of the world are uniform will help with reward learning as well as stabilize the performance across various random seeds.

## Method

To solve the issue of sensitivity of performance to reward initialization, we use our insight as mentioned in section to propose our data-driven reward initialization method which ensures a uniform co-domain of the reward function  $\hat{r}(s, a)$ . Several PbRL works have utilized an unsupervised pre-training step that enables the agent to collect several trajectories in a buffer, from which trajectory pairs are sampled and queried to the user when the main training episodes begin. Say, the agent populates a buffer  $D_\tau$  with trajectories  $\tau_i$  by exploring the world via a policy  $\pi$ . We propose that ensuring that the predicted rewards over all the states visited in all the trajectories  $s_i \in \tau_j \in D_\tau \forall i, j$  are the same, say some non zero value helps the agent learned much faster and ensures that the performance of the learned policy is consistent across various experimental runs (by varying random seeds). Hence, for the acquired dataset  $D_\tau$ , and some non-zero constant  $\epsilon$ , we solve a regression problem by minimizing the following pretraining-initialization loss,

$$\mathcal{L}_P = \sum_{\tau_j \sim D_\tau} \sum_{i=1}^{|\tau_j|} \|R_\psi^h(s_i) - \epsilon\|_2 \quad (3)$$

Note that  $R_\psi^h$  is updated using the pretraining-initialization loss only once to obtain a uniform  $R_\psi^h$ . After this data-driven initialization, we follow the same algorithm as (Lee, Smith, and Abbeel 2021) as our baseline. This additional update to the reward model has zero cost to the human in the loop and a one-time fixed cost for the agent (which is negligible in the context of training both  $R_\psi^h$  and  $\pi$  in a PbRL algorithm.)

## Results

To validate our proposed pretraining-initialization reward model loss, we conduct our experiments on a 7x7 gridworld and a 15x15 gridworld. The gridworld environment is square  $n \times n$  matrix-like domain where the agent starts at the top left corner. The agent can take discrete action steps, such as, Up, Down, Left, and Right to navigate around the gridworld space. The underlying human reward, operationalized via oracular rewards, is assumed such that the goal location is the bottom right corner. We use the negative of euclidean distance from the current agent’s location to the goal location as the underlying human reward function. This oracular reward,  $\tilde{\mathcal{R}}^h$ , is used to provide binary feedbacks when trajectory pairs are queried as:

$$y(\tau_0, \tau_1) = \begin{cases} 0 & \sum_i \tilde{\mathcal{R}}^h(\tau_0) > \sum_i \tilde{\mathcal{R}}^h(\tau_1) \\ 1 & \sum_i \tilde{\mathcal{R}}^h(\tau_0) < \sum_i \tilde{\mathcal{R}}^h(\tau_1) \end{cases} \quad (4)$$

Figure 2 shows the high degree of variability in the initialized reward values for the gridworld states when Xavier (Kumar 2017) weight initialized is used for the reward model. As indicated in previous sections, the predicted rewards are not uniform over the state locations and in fact vary quite significantly across different random seeds. Variability in initial reward estimates is not a problem in itself as the training process is expected to correct for it, however, in practice, we found that in cases when the reward function was initialized “randomly” to align with the goal as in figure 2 (middle) the agent’s performance was significantly better than when the initial reward estimates were incorrect, say, figure 2 (right).

We studied the impact of various popular initialization schemes for deep neural networks such as Kaiming Uniform (He et al. 2015), Xavier Uniform (Kumar 2017), Orthonormal (Hu, Xiao, and Pennington 2020), Zeros and Ones. While initial rewards for the case of zeros weights and ones weight initialization is easy to visualize, figure 3 shows the reward initialization for Xavier (right), Orthonormal (middle), and Ours (left). It can be instantly noticed that in our data-driven initialization, with  $\epsilon = 0.4$ , almost all the states have a predicted reward value of 0.4 which is as expected. However, for Xavier and Orthonormal initialization, the predicted rewards vary significantly across different

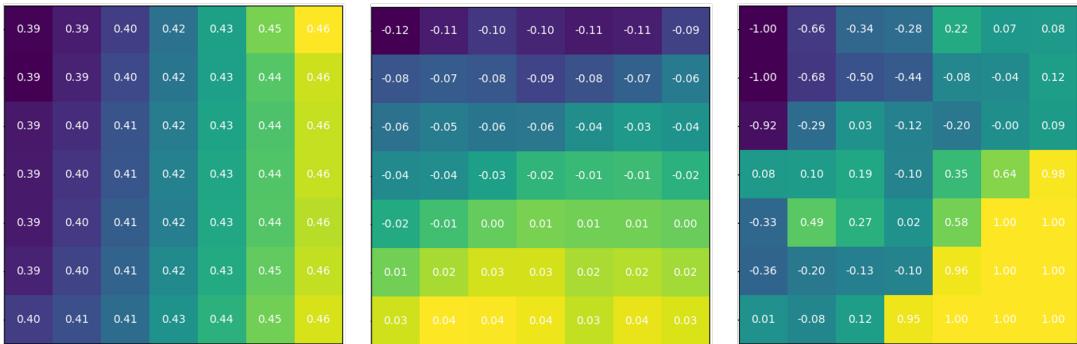


Figure 3: Variations in initial reward values on gridworld states as per Our data-driven approach (left), orthonormal initialization (middle), and Xavier initialization (right). Numbers in each cell show the maximum reward,  $\tilde{\mathcal{R}}^h$ , that the agent upon taking an action in that cell.

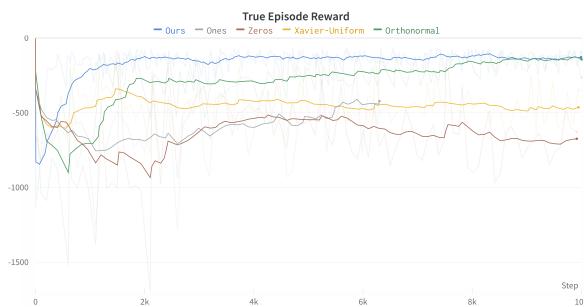


Figure 4: Return of the learned policy on the  $7 \times 7$  gridworld as measured on the ground truth human reward  $\tilde{\mathcal{R}}^h$  comparing Ours with baseline initializations.

states (hence, are “patchy”). We allow 15 episodes of unsupervised pretraining steps for collecting trajectories in the PEBBLE backbone, which doubles as our dataset for minimizing  $\mathcal{L}_P$  loss.

Figure 4 shows the results using various initialization strategies, averaged over 3 independent runs, (Ours, Ones, Zeros, Xavier-Uniform, Orthonormal) on a  $7 \times 7$  gridworld domain with PEBBLE as the backbone PbRL algorithm implemented with DQN (Mnih et al. 2013) as the RL agent (in contrast to SAC (Haarnoja et al. 2018) which works with continuous action spaces), with all hyperparameters being the same across all runs. As expected, the data-driven reward initialization technique proposed in this work has very less variance compared to other initialization techniques (except zeros, where all the predicted rewards are also zeros, but it suffers from poor performance). Additionally, as discussed before, a uniform reward initialization over the state space improves exploration and allows the agent to collect more diverse trajectories thereby improving the queries made to the human in the loop. This also improves the overall performance of Our initialization technique.

We also compare our data-driven initialization with Kaiming-Uniform initialization (He et al. 2015) on a  $15 \times 15$  gridworld with similar semantics as our  $7 \times 7$  gridworld. Figure 5 shows that even in the larger gridworld space our

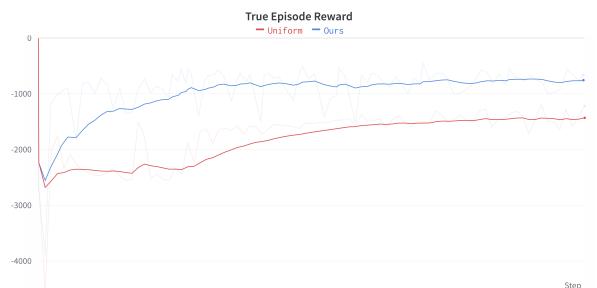


Figure 5: Return of the learned policy on the  $15 \times 15$  gridworld as measured on the ground truth human reward  $\tilde{\mathcal{R}}^h$ .

method can perform better than the baseline with zero additional cost to the human in the loop as well as a negligible cost to the PbRL agent. Figure 1 shows the initial reward values as computed by both the methods Ours (right) and Kaiming-Uniform (left), and further highlights the issue of non-uniform reward values over the state space locations. On the other hand, even though (fig. 1 (right)) reward initialization is not truly uniform, it is more consistent for unseen states (which are farther from the start location of the top-left corner).

## Discussion

In this work, we present the issue of the degeneracy of the reward model in preference-based reinforcement learning works, which in part is impacted by non-uniform predicted rewards over the state space domain. This can cause high variability in the performance measures across various runs and is found to be very sensitive to random seeds. We mitigate this issue by a data-driven reward initialization method that utilizes the states collected while performing unsupervised exploration (a common theme in several PbRL works) to ensure that the predicted rewards over all the states are the same by solving a regression problem. We validate our insights over two gridworld domains of sizes  $7 \times 7$  and  $15 \times 15$  which were chosen to visualize the predicted rewards.

We plan to further investigate the impact of reward ini-

tialization across more complex domains used in PbRL literature like locomotion tasks, robotic manipulation tasks, and other explicit discrete action domains. Additionally, this work calls for a more thorough investigation of potential reasons that worsen the issue of reward function degeneracy. Finally, we also plan to study the impact of reward initialization across various PbRL methods like (Park et al. 2022; Verma and Metcalf 2022; Christiano et al. 2017).

## Acknowledgements

Kambhampati's research is supported by the J.P. Morgan Faculty Research Award, ONR grants N00014-16-1-2892, N00014-18-1-2442, N00014-18-1-2840, N00014-9-1-2119, AFOSR grant FA9550-18-1-0067 and DARPA SAIL-ON grant W911NF19-2-0006.

## References

- Arulkumaran, K.; Deisenroth, M. P.; Brundage, M.; and Bharath, A. A. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6): 26–38.
- Bishop, C. M.; and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.
- Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.
- Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.
- Gopalakrishnan, S.; Verma, M.; and Kambhampati, S. 2021a. Computing Policies That Account For The Effects Of Human Agent Uncertainty During Execution In Markov Decision Processes. *arXiv preprint arXiv:2109.07436*.
- Gopalakrishnan, S.; Verma, M.; and Kambhampati, S. 2021b. Synthesizing Policies That Account For Human Execution Errors Caused By State Aliasing In Markov Decision Processes. In *ICAPS 2021 Workshop on Explainable AI Planning URL <https://openreview.net/pdf>*.
- Guan, L.; Verma, M.; Guo, S. S.; Zhang, R.; and Kambhampati, S. 2021. Widening the pipeline in human-guided reinforcement learning with explanation and context-aware data augmentation. *Advances in Neural Information Processing Systems*, 34: 21885–21897.
- Guan, L.; Verma, M.; and Kambhampati, S. 2020. Explanation augmented feedback in human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2006.14804*.
- Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.
- Hadfield-Menell, D.; Milli, S.; Abbeel, P.; Russell, S. J.; and Dragan, A. 2017. Inverse reward design. *Advances in neural information processing systems*, 30.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.
- Hu, W.; Xiao, L.; and Pennington, J. 2020. Provable benefit of orthogonal initialization in optimizing deep linear networks. *arXiv preprint arXiv:2001.05992*.
- Hussein, A.; Gaber, M. M.; Elyan, E.; and Jayne, C. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2): 1–35.
- Kambhampati, S.; Sreedharan, S.; Verma, M.; Zha, Y.; and Guan, L. 2022. Symbols as a lingua franca for bridging human-ai chasm for explainable and advisable ai systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 12262–12267.
- Krakovna, V.; Uesato, J.; Mikulik, V.; et al. 2020. Specification gaming: The flip side of AI ingenuity—DeepMind.
- Kumar, S. K. 2017. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*.
- Lee, K.; Smith, L.; and Abbeel, P. 2021. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*.
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- 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. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.
- Narkhede, M. V.; Bartakke, P. P.; and Sutaone, M. S. 2022. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, 55(1): 291–322.
- Ng, A. Y.; Russell, S.; et al. 2000. Algorithms for inverse reinforcement learning. In *Icmi*, volume 1, 2.
- Park, J.; Seo, Y.; Shin, J.; Lee, H.; Abbeel, P.; and Lee, K. 2022. SURF: Semi-supervised Reward Learning with Data Augmentation for Feedback-efficient Preference-based Reinforcement Learning. *arXiv preprint arXiv:2203.10050*.
- Schaal, S. 1996. Learning from demonstration. *Advances in neural information processing systems*, 9.
- Shah, R.; Freire, P.; Alex, N.; Freedman, R.; Krasheninnikov, D.; Chan, L.; Dennis, M. D.; Abbeel, P.; Dragan, A.; and Russell, S. 2020. Benefits of assistance over reward learning.
- Soni, U.; Sreedharan, S.; Verma, M.; Guan, L.; Marquez, M.; and Kambhampati, S. 2022. Towards customizable reinforcement learning agents: Enabling preference specification through online vocabulary expansion. *arXiv preprint arXiv:2210.15096*.
- Sreedharan, S.; Soni, U.; Verma, M.; Srivastava, S.; and Kambhampati, S. 2020. Bridging the Gap: Providing Post-Hoc Symbolic Explanations for Sequential Decision-Making Problems with Inscrutable Representations. *arXiv preprint arXiv:2002.01080*.
- Vamplew, P.; Dazeley, R.; Foale, C.; Firmin, S.; and Mummary, J. 2018. Human-aligned artificial intelligence is a

multiobjective problem. *Ethics and Information Technology*, 20(1): 27–40.

Verma, M.; Bhambri, S.; Gupta, S.; and Buduru, A. B. 2019. Making Smart Homes Smarter: Optimizing Energy Consumption with Human in the Loop. *arXiv preprint arXiv:1912.03298*.

Verma, M.; Kharkwal, A.; and Kambhampati, S. 2022. Advice Conformance Verification by Reinforcement Learning agents for Human-in-the-Loop. *arXiv preprint arXiv:2210.03455*.

Verma, M.; and Metcalf, K. 2022. Symbol Guided Hindsight Priors for Reward Learning from Human Preferences. *arXiv preprint arXiv:2210.09151*.

Verma, M.; Shah, N.; Nayyar, R. K.; and Hanni, A. 2021. Perfect Observability is a Myth: Restraining Bolts in the Real World.

Zahedi, Z.; Sreedharan, S.; Verma, M.; and Kambhampati, S. 2022. Modeling the Interplay between Human Trust and Monitoring. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 1119–1123. IEEE.

Zahedi, Z.; Verma, M.; Sreedharan, S.; and Kambhampati, S. 2021. Trust-aware planning: Modeling trust evolution in longitudinal human-robot interaction. *arXiv preprint arXiv:2105.01220*.