PHIRL: Progress Heuristic for Inverse Reinforcement Learning

Hang Yu, James Staley, Shijie Fang, Wenchang Gao, Reuben M. Aronson, and Elaine S. Short
School of Engineering
Tufts University
Medford, MA 02155, USA

***Abstract*—In this paper, we propose a new framework, PHIRL, leveraging both human demonstrations and a recently introduced type of human feedback, *progress*. *Progress* describes the completion rate of a task based on the end state of a trajectory and has been shown to correlate with task success while being consistent across multiple non-experts. We use *progress* to annotate a part of the demonstration dataset. In PHIRL, reward functions are learned using IRL methods and then shaped to align with the *progress* annotations over the annotated demonstrations. Our method does not intensively rely on humans to stay in the learning loop to provide feedback during the training and is capable of mitigating reward hacking and bottleneck issues. We validate PHIRL using a simulation study and a block lifting task. Our results show that PHIRL learns better reward functions and is more robust when the demonstrations are imperfect.**

I. Introduction

Reinforcement Learning (RL) has become one of the most popular methods for enabling robots to learn new skills [25, 44]. An essential component for RL agents to perform well is reward functions. In most cases, reward functions are often hard-coded by system designers [30]. Inverse Reinforcement Learning (IRL) offers an alternative by inferring the latent reward function R directly from human demonstrations. IRL often requires human demonstrations to be optimal, or the majority of demonstrations are near-optimal [36]. Prior work indicated that using human feedback along with human demonstrations can relax the quality constraints for demonstrations. In this work, we focus on effectively learning robust reward functions by leveraging human demonstrations and a novel teaching signal *progress*.

Providing high-quality to manipulation tasks is known to be challenging even for experts. Nevertheless, high-quality human feedback is more accessible since providing feedback requires less mental effort and skills compared to providing demonstrations. Most recent work has demonstrated that using human demonstrations and human feedback jointly can significantly improve data efficiency since human feedback and human demonstration have complementary advantages [30]. Human demonstrations contain dense information, require more skills to provide, and often have some suboptimality. Human feedback offers relatively sparse information, but requires less expertise and tends to be more precise. Thus, using human feedback as supplementary for reward learning, or using human demonstrations to boost feedback collection, can lead to more effective and robust learning.

Despite the improvement in learning by combining human feedback and human demonstrations is significant, the complementary advantages are not fully utilized in prior work [30, 20, 6]. The human demonstrations are generally used to pre-train robots to more efficiently sample human feedback, and then discarded in the later part of learning. Human feedback is often used to indicate users' preferences. However, if generated trajectories are both good or bad, forcing users to pick one preferred trajectory would not benefit learning , and users are required to stay in the training loop to provide feedback to newly generated trajectories. Thus, there is a need for a more effective reward-learning method to further reduce human efforts in the training loop while learning an accurate reward function. **Our key intuition** is that: *when humans are solving a problem, we are not only inferring the answers and then asking for evaluations but also validating our inferences on instances with known answers.*

We use this insight to develop an efficient reward learning framework, PHIRL, that learns from both demonstrations and a recently introduced type of human feedback, *progress*. *Progress* is a signal that describes the degree of task completion. We use *progress* to annotate a part of human demonstration data. We first use an IRL algorithm to learn a reward function from all demonstrations, and then shape the learned reward function to be consistent with the *progress*. Our method is: (1) more robust to noisy or even failed demonstrations since low-quality demonstrations would receive low or negative progress and thus small or negative rewards; (2) more robust to reward hacking since reward hacking behaviors are not making actual progress; (3) more flexible in human participation levels since the annotations can be collected before learning. We validate our framework on a simulated robot in Robosuite with Robomimic datasets. Our results showed that PHIRL is significantly more effective and robust than the baseline with low-quality and high-quality demonstrations.

II. Background

Interactive machine learning grants robots the ability to effectively adapt to human needs or learn new skills by leveraging human knowledge [3]. There are diverse representations of human knowledge, such as human preference [2], verbal feedback [28], eye gaze [35], facial expressions [12], numerical evaluation [23], and human demonstrations [33]. Each representation has its advantages and limitations. One

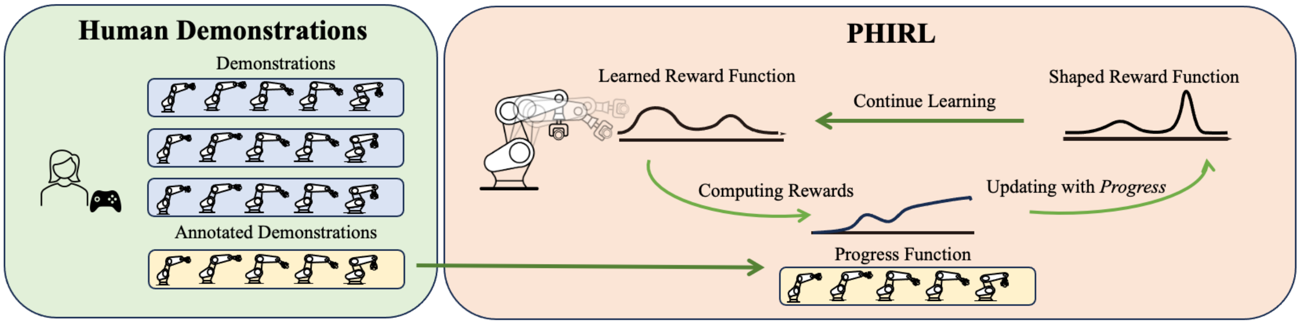

Fig. 1. The PHIRL framework. Human users first provide demonstrations. Then we randomly sample a subset of demonstrations and ask human users to annotate the demonstrations with *progress*. We then alternately use an IRL algorithm to learn a reward function from the demonstrations and align the learned reward function to be consistent with the information carried by *progress* annotations. This process will be repeated until the reward learner achieves satisfied performance. Optionally, the trajectories generated by the robot during learning could be added to the annotated demonstration dataset after being annotated to further improve learning.

approach to achieve better learning is to use multiple forms of human knowledge jointly [30]. In this work, we seek to improve Inverse Reinforcement Learning by integrating human feedback and human demonstrations.

Learning from Human Feedback Human feedback is a popular form of human teaching signals for robots to learn from humans via interactive reinforcement learning or inverse reinforcement learning [1, 9, 24, 8, 26, 22, 15, 12]. To provide human feedback, humans first observe robots performing a task, and then evaluate the performance of the robots implicitly or explicitly [3]. Humans can use feedback to assess robot behaviors [40] and indicate their preferred policy [2]. In our work, we choose an effective and easy-to-provide signal *progress* [43] and reduce the exploration problem by combining *progress* with human demonstrations.

Learning from Demonstrations Learning from Demonstrations (LfD) allows robots to learn to perform new tasks by imitating humans [38, 26, 29, 4, 31, 10, 36, 41]. LfD approaches have many advantages, such as reducing the needs for expert programming [46], data efficiency [32], and guaranteed task performance[21]. LfD generally solves the learning problem in two major ways: (1) inferring a policy[16, 11, 18, 19], and (2) inferring a reward function[34, 17, 42, 13]. Methods that infer policies, such as DreamerV3 and Diffusion Policy [16, 11], have higher data efficiency and might be more robust, while methods that infer reward functions, such as MaxEnt IRL and AIRL [47, 13], better interpret human's intentions. With sufficient and error-free demonstrations, LfD methods are guaranteed to produce optimal behaviors [37]. Despite LfD methods have achieved significant improvements in the past few years in learning from non-perfect demonstrations [36, 39], the majority of demonstrations still need to be relatively high-quality and consistent. Especially for IRL, learning an accurate reward function purely from demonstrations is challenging [30].

Combining LfD and LfHF Recent work indicates that combining human demonstrations and human feedback could improve learning efficiency and overcome the limitation of using human feedback or demonstrations solely [20, 5, 30, 26, 7]. Specifically, Ibarz et al. [20] proposed to use demonstrations to pre-train the learning agent, and then used the pre-train agent to efficiently generate meaningful preference queries, which significantly improves the sample efficiency for preference learning. Building on Ibarz et al. [20]'s work, Palan et al. [30] not only used the demonstrations to improve sample efficiency but also used the demonstrations to learn an initial reward model. Although prior work has successfully consolidated human demonstration with human feedback, humans still need to stay in the training loop to continuously provide feedback.

Our work differs from prior work by focusing on learning robust and accurate reward functions and improving data efficiency. The learned reward functions are aligned with human *progress* annotations, which makes the learning outcome more robust to reward hacking and incomplete demonstrations.

III. Methodology

We aim to learn a robust function effectively from both human demonstrations and human feedback without requiring humans to be in the learning loop. Previous work has demonstrated that *progress* is an informative type of human feedback and has many advantages when used to annotate demonstrations, especially when annotators are multiple non-experts. Our **key insights** are:

> *Reward functions are used to drive agents to complete the task, and progress describes how much of a task has been completed.*

We then develop our novel reward learning framework, PHIRL, by aligning machine-learned information and human knowledge on the same demonstration dataset.

A. Progress

To effectively annotate human demonstrations, we use a novel human feedback form, *progress*. Progress is a signal that describes the accumulative degree of task completion, ranging from fully incomplete to fully complete. Progress can be collected by asking users to observe a single state or a trajectory. Previous work [43] has demonstrated that progress is an informative teaching signal and has many advantages for reward learning. Progress indicates whether the robot is acting toward task completion in a scale value and if the task has been completed. The progress signal we used ranges from 0 to 100, where 100 means the task is perfectly complete and 0 means there is no progress has been made yet. The value of progress depends on how far along the task has been completed at a single state or the end state of a trajectory, and is relatively independent from state transitions. Moreover, progress is robust to non-expert demonstrations and more consistent across non-experts, which allows us to collect both demonstrations and progress from a broader source. In this work, we use *progress* as the reference for reward function alignment.

B. Reward Learning from Demonstrations

Given a set of human demonstrations $D = \{d_0, ..., d_n\}$, where $d = (s_0, a_0, ..., s_n, a_n)$, the objective of Inverse Reinforcement Learning (IRL) is to find a reward function $r(s, a)$. To learn robust reward functions efficiently, PHIRL alternately learns a reward function and shapes it to align with *progress*. If we assume that the demonstrations are from an optimal policy π^*, we can then interpret the IRL problem as a maximum likelihood problem:

$$\max_{\theta} E_{d\sim D}[\log p_\theta(d)] \tag{1}$$

where

$$p_\theta(d) \propto p(s_0) \prod_{t=0}^{T} p(s_{t+1}|s_t, a_t) e^{\gamma t r_\theta(s_t, a_t)}$$

parametrizes the reward function. We choose the Adversarial Inverse Reinforcement Learning (AIRL) [13] algorithm to learn the initial reward distribution. AIRL casts the optimization of Equation 1 as a GAN [14] optimization problem. The discriminator uses a particular f_θ:

$$D_\theta(s, a, s_0) = \frac{\exp\{f_\theta(s, a, s_0)\}}{\exp\{f_\theta(s, a, s_0)\} + \pi(a|s)} \tag{2}$$

and π is trained to maximize:

$$r(d) = \log(1 - D(d)) - \log D(d) \tag{3}$$

, where $f_\theta(s, a, s_0)$ can be interpret as the advantage under deterministic dynamics:

$$f^*(s, a, s') = \underbrace{r^*(s) + \gamma V^*(s')}_{Q(s,a)} - \underbrace{V^*(s)}_{V(s)} = A^*(s, a) \tag{4}$$

We choose AIRL as our reward learning method since it learns a state-only value function, which aligns with the intuition of *progress*. Similar to *progress* at state s, the learning outcome of AIRL consists of $r(s)$ and $V(s)$, which are both state-only functions.

C. Reward Function Shaping

Our goal is to improve the reward functions learned from human demonstrations using progress. Specifically, we exploit four key aspects of the information provided by *progress* to shape the learned rewards: 1) positive *progress* indicates positive rewards 2) larger increase in *progress* indicates larger rewards; 3) high current *progress* indicates high potential; 4) successful demonstrations should receive more rewards than incomplete/failed demonstrations.

Positive delta *progress*, positive rewards. Given a trajectory segment $s_1 \xrightarrow{a_s} s_2$, if the robot's action sequence is appropriate, the *progress* at state s_2 should be higher at s_1. Similarly, the learned incremental reward $r(s_1, s_2)$ should also be positive. We enforce this relationship via the binary cross-entropy (BCE) loss:

$$\mathcal{L}(\Delta p \to r) = \text{BCE}\left(\sigma\left(p(s_1, s_2)\right), \sigma\left(r(s_1, s_2)\right)\right) \tag{5}$$

where σ denotes the sigmoid function, $p(s_1, s_2) = p(s_2) - p(s_1)$.

More increase in *progress*, more rewards. Even in optimal demonstrations, not all steps in a demonstration are equally important. Some steps, like picking behaviors in a pick-and-place task, are inherently more important than other steps, and intuitively should receive a larger reward than other steps. If a state or a sub-trajectory has a higher delta progress than another state or sub-trajectory, the learned reward should also be higher. We enforce this by:

$$\begin{aligned}\mathcal{L}(\Delta\Delta p \to \Delta r) = \text{BCE}(&p(s_1, s_2) - p(s_3, s_4), \\ &r(s_1, s_2) - r(s_3, s_4))\end{aligned} \tag{6}$$

Noted, in this condition specifically, $s_1 \xrightarrow{a_s} s_2, s_3 \xrightarrow{a_s} s_4$, and s_1, s_2, s_3, and s_4 are from the same demonstration since different users might scale *progress* differently.

High *progress*, high potential. The potential function Φ was introduced in [27], and describes the distance between the current state and the goal state (i.e. $\Phi(s) = -dist(s, s_{goal})$. The use of potential function will not alter the original optimal policy, and the agent can reduce the need for random exploration by getting heuristics from $\gamma\Phi(s') - \Phi(s)$. If a state or a sub-trajectory has higher progress than another state or sub-trajectory, the learned potential should also be higher. We captured this by:

$$\mathcal{L}(\Delta p \to \Phi) = \text{BCE}(p(s_1, s_2), \Phi(s_1, s_2)) \tag{7}$$

Final Status Progress and Total Reward Rankings The ultimate goal of reward learning is to learn a reward function that guides the agent to successfully complete the task. Therefore, any demonstration that completed the task should receive a higher total reward than any demonstration that did not complete the task. Previous work has indicated that most failed demonstrations have a *progress* that is lower than 90 at the end of the demonstration. In this work, we use 90 to distinguish between successful and failed demonstrations. We

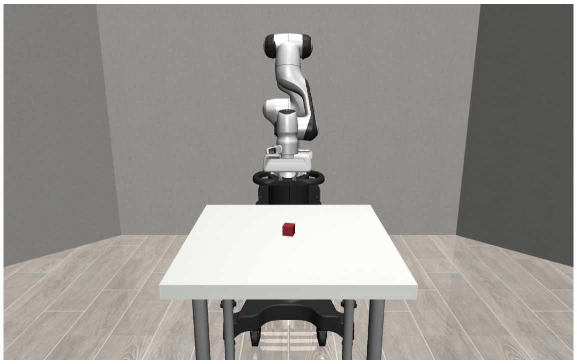

Fig. 2. Simulation Task Environment. Lift: the arm needs to reach the block, grasp the block, and lift the block from the table.

capture this condition by:

$$\mathcal{L}(p \rightarrow ranking) = \sum_{d_f \in D_f} \left(R(d_f) - \min_{d_s \in D_s} (R(d_s)) \right) \quad (8)$$

where D_s are all successful demonstrations in the demonstration set D and D_f are all failed or incomplete demonstrations in D.

D. PHIRL

We showed the general workflow of our learning framework in Algorithm 1. All demonstrations and annotations are collected before the learning phase. Progress annotations are used to shape the learned reward function to boost training and stabilize the learning outcome. The method would benefit from human-in-the-loop for improving robustness, but human-in-the-loop teaching is mandatory. The procedure for learning a reward function will have the following steps, which will be described later in the section:

Algorithm 1 PHIRL

```
Collect demonstration dataset $\mathcal{D}$
Sample a subset of demonstrations $\mathcal{D}_p$ from $\mathcal{D}$
For trajectory $\tau \in$ demonstration $d$ and $d \in \mathcal{D}_p$, annotate $\tau$ with progress $p$
while not converged do
    Sample trajectories $\tau_{\mathcal{D}}$ from $\mathcal{D}$ and $\tau_\pi$ from policy $\pi$
    Train $D_{\theta,\phi}$ by distinguishing $\tau_{\mathcal{D}}$ and $\tau_\pi$
    Sample trajectories $\tau_{\mathcal{D}_p}$ from $\mathcal{D}_p$ and corresponding progress labels $p_\tau$
    Update $D_{\theta,\phi}$ by aligning $r_{\theta,\phi}(\tau_{\mathcal{D}_p})$ with $p_\tau$, where
    $r_{\theta,\phi}(s,a,s') \leftarrow \log\big(D_{\theta,\phi}(s,a,s')\big) - \log\big(1 - D_{\theta,\phi}(s,a,s')\big)$.
    Optimizing policy $\pi$ respect to updated $r_{\theta,\phi}$
    Optional: active sampling and querying
      • Sample a demonstration $d_\pi$ from policy $\pi$
      • Annotate $d_\pi$ with progress, $\mathcal{D}_p \cup d_\pi$
    Check for Convergence
end while
```

E. Online Updating and Scalability

Optionally, we included an online update process in PHIRL. While human-in-the-loop teaching is not required for our method, PHIRL would benefit from adding generated demonstrations into the annotation dataset, especially if the generated demonstration has unexpected failures, such as repetitively picking and releasing the target object or performing a pick and place task without actually picking the object up. This will further improve the robustness of the learning outcome, and align the reward function with actual human intentions.

We choose AIRL over other IRL algorithms because we believe that AIRL's training objectives are well-aligned with progress in multiple aspects, but the learning framework is not necessarily bonded with AIRL. The key idea of the framework is shaping the learned rewards to align with the evaluation information specified by progress. Most types of shaping loss proposed in this paper do not require the learning algorithm to be AIRL. We will further demonstrate this with our experiment results.

IV. Experiments and Results

In this section, we will first talk about simulation experiment setups, and then discuss the results. To analyze data, we used Welch's t-test to indicate significant in our results.

A. Task And Dataset

The simulation environment we used in this paper is Robosuite. [45]. We selected Lift from Robosuite, where the robot arm only needs to reach the target block and pick it up to an arbitrary height. The tasks are shown in **??**. We trained PHIRL on a high-quality dataset and a mixed-quality dataset from Robomimic [] separately. The high-quality dataset is the PH dataset, which is provided by one single proficient operator using the RoboTurk platform. The dataset consists of 200 successful demonstrations. The mixed-quality dataset is the MH dataset, which is collected by six operators using the RoboTurk platform. Each operator provided 50 demonstrations and each operator has varied proficiency. Two operators were "worse" operators, two were "okay" operators, and two were "better" operators, resulting in a mixed-quality dataset. For each dataset, we annotated ten percent of the demonstrations. Each demonstration was equally divided into 10 sub-trajectories, and each sub-trajectory was annotated with a progress label. All annotations were done by three robot experts with limited experience with Robosuite.

B. Results

We used a fixed seed, picked the best AIRL and PHIRL models in 50 million steps, and ran the agents in the environment for 50 rounds over 300 steps. We showed results for the Lift task in subsection IV-A. The success rates for agents learned from AIRL and PHIRL are shown on the left. PHIRL achieved largely more success than AIRL on both datasets (32 V.S. 5 on ph, and 11 V.S. 0 on mh). PHIRL also obtained significantly higher average rewards on two datasets. The average environmental rewards for the agents that

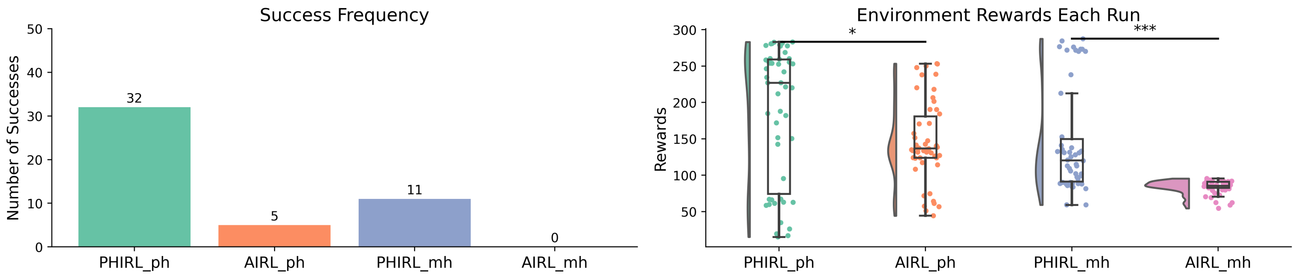

Fig. 3. PHIRL Versus AIRL on Lift Task. In 50 runs, PHIRL achieved more success and higher average rewards with both datasets. The difference is more significant on the mh dataset, where the demonstration quality is lower than the ph dataset.

learned ph dataset are: 186.42 (PHIRL) and 146.11 (AIRL), $t = 2.63, p < 0.001$. The average environmental rewards for the agents that learned mh dataset are: 146.82 (PHIRL) and 83.93 (AIRL), $t = 5.90, p < 0.001$.

V. Discussion and Conclusion

We demonstrated that PHIRL learns better reward functions than AIRL on a lifting task and is more robust when demonstrations are imperfect, while only requiring to annotate 10% of collected demonstrations before the learning phase. One limitation of this work is that we claimed that PHIRL would benefit from having humans in the loop, but we did not test and demonstrate the effect of using PHIRL in an online setting. In our future work, we will conduct a human study with a real robot over a long-horizon task. We will collect human feedback and human demonstrations from non-experts and test PHIRL in both online and offline feedback.

In conclusion, in this work, we proposed PHIRL, an Inverse Reinforcement Learning method that learns reward functions by alternately learning a reward function and shaping the learned reward function with *progress*. We showed that PHIRL significantly outperforms AIRL in a block lifting task with a higher successful rate on both the perfect human and the mixture human demonstration datasets.

References

[1] Learning from feedback on actions past and intended, author=Knox, W Bradley and Breazeal, Cynthia and Stone, Peter. In *In Proc. of 7th ACM/IEEE Intl. Conf. on Human-Robot Interaction, Late-Breaking Reports Session (HRI 2012)*, 2012.

[2] Riad Akrour, Marc Schoenauer, and Michele Sebag. Preference-based policy learning. In *Machine Learning and Knowledge Discovery in Databases: European Conf., ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proc., Part I 11*, pages 12–27. Springer, 2011.

[3] Christian Arzate Cruz and Takeo Igarashi. A survey on interactive reinforcement learning: Design principles and open challenges. In *Proc. of the 2020 ACM designing interactive systems Conf.*, pages 1195–1209, 2020.

[4] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *Int. Conf. on machine learning*, pages 783–792. PMLR, 2019.

[5] Daniel Brown, Russell Coleman, Ravi Srinivasan, and Scott Niekum. Safe imitation learning via fast bayesian reward inference from preferences. In *Int. Conf. on Machine Learning*, pages 1165–1177. PMLR, 2020.

[6] Daniel S Brown, Wonjoon Goo, and Scott Niekum. Ranking-based reward extrapolation without rankings. *arXiv preprint arXiv:1907.03976*, 2019.

[7] Daniel S Brown, Wonjoon Goo, and Scott Niekum. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conf. on robot learning*, pages 330–359. PMLR, 2020.

[8] Kate Candon, Nicholas C Georgiou, Helen Zhou, Sidney Richardson, Qiping Zhang, Brian Scassellati, and Marynel Vázquez. React: Two datasets for analyzing both human reactions and evaluative feedback to robots over time. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, pages 885–889, 2024.

[9] Thomas Cederborg, Ishaan Grover, Charles L Isbell Jr, and Andrea Lockerd Thomaz. Policy shaping with human teachers. In *IJCAI*, pages 3366–3372, 2015.

[10] Letian Chen, Rohan Paleja, and Matthew Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. In *Conf. on robot learning*, pages 1262–1277. PMLR, 2021.

[11] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[12] Yuchen Cui, Qiping Zhang, Brad Knox, Alessandro Allievi, Peter Stone, and Scott Niekum. The empathic framework for task learning from implicit human feedback. In *Conf. on Robot Learning*, pages 604–626. PMLR, 2021.

[13] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning, 2018.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks.

Communications of the ACM, 63(11):139–144, 2020.

[15] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. In *Adv. in neural information processing Sys.*, pages 2625–2633, 2013.

[16] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[17] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[18] Shengran Hu and Jeff Clune. Thought cloning: Learning to think while acting by imitating human thinking. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Mostafa Hussein, Brendan Crowe, Madison Clark-Turner, Paul Gesel, Marek Petrik, and Momotaz Begum. Robust behavior cloning with adversarial demonstration detection. In *2021 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 7858–7864. IEEE, 2021.

[20] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

[21] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[22] W Bradley Knox and Peter Stone. Tamer: Training an agent manually via evaluative reinforcement. In *2008 7th IEEE Intl. Conf. on Development and Learning*, pages 292–297. IEEE, 2008.

[23] W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proc. of the fifth Int. Conf. on Knowledge capture*, pages 9–16, 2009.

[24] Jacky Liang, Fei Xia, Wenhao Yu, Andy Zeng, Montserrat Gonzalez Arenas, Maria Attarian, Maria Bauza, Matthew Bennice, Alex Bewley, Adil Dostmohamed, et al. Learning to learn faster from human feedback with language model predictive control. *arXiv preprint arXiv:2402.11450*, 2024.

[25] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[26] Shaunak A Mehta and Dylan P Losey. Unified learning from demonstrations, corrections, and preferences during physical human–robot interaction. *ACM Transactions on Human-Robot Interaction*, 13(3):1–25, 2024.

[27] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[28] Stefanos Nikolaidis, Minae Kwon, Jodi Forlizzi, and Siddhartha Srinivasa. Planning with verbal communication for human-robot collaboration. *ACM Transactions on Human-Robot Interaction (THRI)*, 7(3):1–21, 2018.

[29] Tao Ning, Chunhong Zhang, Zheng Hu, Xiaosheng Tang, and Benhui Zhuang. Learning from imperfect demonstrations via reweighting confidence. In *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*, pages 1105–1112. IEEE, 2022.

[30] Malayandi Palan, Nicholas C Landolfi, Gleb Shevchuk, and Dorsa Sadigh. Learning reward functions by integrating human demonstrations and preferences. *arXiv preprint arXiv:1906.08928*, 2019.

[31] Georgiy Pshikhachev, Dmitry Ivanov, Vladimir Egorov, and Aleksei Shpilman. Self-imitation learning from demonstrations. *arXiv preprint arXiv:2203.10905*, 2022.

[32] Harish Ravichandar, S Reza Ahmadzadeh, M Asif Rana, and Sonia Chernova. Skill acquisition via automated multi-coordinate cost balancing. In *2019 Int. Conf. on Robotics and Automation (ICRA)*, pages 7776–7782. IEEE, 2019.

[33] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3:297–330, 2020.

[34] Juntao Ren, Gokul Swamy, Zhiwei Steven Wu, J Andrew Bagnell, and Sanjiban Choudhury. Hybrid inverse reinforcement learning. *arXiv preprint arXiv:2402.08848*, 2024.

[35] Akanksha Saran, Elaine Schaertl Short, Andrea Thomaz, and Scott Niekum. Understanding teacher gaze patterns for robot learning. In *Conf. on Robot Learning*, pages 1247–1258. PMLR, 2020.

[36] Fumihiro Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *Int. Conf. on Learning Representations*, 2020.

[37] Fumihiro Sasaki, Tetsuya Yohira, and Atsuo Kawaguchi. Sample efficient imitation learning for continuous control. In *Int. Conf. on learning representations*, 2018.

[38] Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

[39] Yunke Wang, Chang Xu, Bo Du, and Honglak Lee. Learning to weight imperfect demonstrations. In *Int. Conf. on Machine Learning*, pages 10961–10970. PMLR, 2021.

[40] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. *arXiv preprint arXiv:1709.10163*, 2017.

[41] Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *Int. Conf. on Machine Learning*, pages 6818–6827. PMLR, 2019.

[42] Biao Yang, Yanan Lu, Rui Wan, Hongyu Hu, Changchun Yang, and Rongrong Ni. Meta-irlsot++: A meta-inverse reinforcement learning method for fast adaptation of trajectory prediction networks. *Expert Systems with Applications*, 240:122499, 2024.

[43] Hang Yu, Qidi Fang, Shijie Fang, Reuben M Aronson, and Elaine Schaertl Short. How much progress did i make? an unexplored human feedback signal for teaching robots. In *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, pages 1739–1746. IEEE, 2024.

[44] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2019. URL https://arxiv.org/abs/1910.10897.

[45] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning, 2025. URL https://arxiv.org/abs/2009.12293.

[46] Zuyuan Zhu and Huosheng Hu. Robot learning from demonstration in robotic assembly: A survey. *Robotics*, 7(2):17, 2018.

[47] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.