Reward Regression from Varying Levels of Human Suboptimal Demonstrations

Abivishaq Balasubramanian, Arjun Rahar, and Jung Ho Yoo
School of Interactive Computing, Aerospace Engineering, and Electrical and Computer Engineering
abivishaq@gatech.edu, arahar3@gatech.edu, and jyoo352@gatech.edu

Abstract: In robotics, the efficacy of Learning from Demonstration (LfD) is often limited by the quality of human demonstrations, which are typically suboptimal. Previous approaches [1] address suboptimality by injecting noise into demonstrations to simulate performance degradation. Observing the trend between the rewards corresponding with degrading performance allows regressing to find a reward closer to thes true reward. This work challenges the assumption made about the relationship between the suboptimal demonstration and an optimal one. Varying levels of distraction are collected to create a more realistic spectrum of suboptimal demonstrations by humans. By integrating distraction as a variable, we explore its impact on the demonstrator's performance in tasks such as "CartPole," "Pendulum", "Frozen Lake," and "Car Racing" within gym environments. Final results indicate that using varying levels of human suboptimal demonstration not only enhances the understanding of performance degradation factors but also leads to the derivation of better-performing policies compared to traditional noise-based methods. This research could significantly enhance the adaptability and efficiency of autonomous systems in learning new tasks through non-expert human demonstrations.

1 Introduction

Learning from Demonstration (LfD) is a powerful paradigm in robotics that enables robots to acquire new skills or tasks by learning from human behavior. This method is particularly advantageous in complex environments where programming explicit instructions is impractical or impossible. However, the effectiveness of LfD is significantly constrained by the quality of the demonstrations provided [2]. Typically, human demonstrations are inherently suboptimal due to various factors such as lack of expertise, inconsistency in task performance, or physical limitations, which pose challenges in deriving optimal robotic policies.

However, existing work in Inverse Reinforcement Learning (IRL) often depends on assumptions that are only valid in very controlled or isolated situations [3]. For instance, Maximum Margin IRL [4] presumes that demonstrations are optimal, which limits its ability to enhance performance when faced with suboptimal demonstrations. Other probabilistic approaches to IRL, such as Maximum Entropy IRL [5] and Bayesian IRL [6], introduce the concept of stochastic or soft optimality, which allows for some deviation from perfect optimality. Although these probabilistic models are capable of identifying the optimal rewards and policies from demonstrations that are slightly suboptimal under these relaxed assumptions, they generally struggle to significantly improve upon the policies derived from suboptimal demonstrations.

Traditional approaches to LfD involve the use of expert demonstrations or the application of noise to simulate suboptimal performance, aiming to understand and compensate for the deviation from optimal behavior [1]. These methods rely on the assumption that noise can adequately represent the variability and errors inherent in human demonstrations [7]. Despite their utility, these approaches often fail to capture the complex, dynamic nature of human suboptimality, leading to subpar policy learning.

In contrast to noise-based methods, this project introduces an innovative approach by using varying levels of distraction to induce suboptimal performances more naturally. Distraction here is defined as any external factor that diverts the demonstrator's attention away from performing the task optimally. This method is grounded in the hypothesis that distractions more accurately reflect real-world conditions that affect human performance. By integrating real-life variables such as distractions into LfD, we aim to model human error and performance variability more realistically, which could significantly enhance the fidelity of the learned policies.

This research is motivated by the potential of distraction-based LfD to not only improve the accuracy of robot learning from non-expert demonstrations but also to increase the accessibility of robotic systems to users who may not have the skills to provide perfect demonstrations. For example, in a domestic setting, a robot trained via this method could effectively learn household tasks from demonstrations performed under various levels of distraction, reflecting typical home environments.

The following sections will detail the methodologies employed, the experimental setup, results from our tests, and discussions on the findings.

2 Related Works

The challenge of Learning from Demonstration (LfD) in robotics often revolves around the optimization of policies from suboptimal human demonstrations. Traditional methodologies typically rely on injecting noise into demonstrations to simulate various levels of demonstrator performance. This section reviews seminal works in the field that have influenced the current project, highlighting the transition from noise-based methods to our novel distraction-based approach.

Noise-Induced Suboptimality: Brown et al. [7] introduced a method to improve imitation learning by automatically ranking demonstrations based on the induced noise levels, enabling the regression to a reward function closer to the demonstrator's true intent [7]. Similarly, Chen et al. [1] enhanced this approach by adding noise directly to the trajectories rather than through a behavior-cloned policy, suggesting that direct manipulation of the demonstration trajectory yields more effective learning outcomes [1]. These studies provide a foundational understanding of how artificially induced suboptimality can facilitate more robust LfD.

Probabilistic and Bayesian Methods: Beyond simple noise addition, probabilistic approaches like Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) introduced by Ziebart et al. [5], use the entropy of the policy to capture the variability in human behavior, thereby allowing for stochastic deviations from optimality. Bayesian IRL (BIRL), developed by Ramachandran and Amir [6], extends this by providing a probabilistic framework that infers the underlying reward functions based on observed behavior, accommodating uncertainties inherent in human demonstrations.

Preference-Based and Ranking Approaches: Recent methods have also explored the use of preference-based learning where demonstrations are not directly used to learn actions but to learn preferences or rankings among actions. Wirth et al. [8] presented an approach that learns policies based on user preferences instead of attempting to replicate demonstration trajectories, adapting to the quality of demonstrations.

Critiques and Limitations: While noise addition has been proven useful, it is predicated on the assumption that the impact of noise on demonstration quality is analogous to naturally occurring suboptimal performance, which may not always hold true. This assumption can lead to discrepancies between the learned and optimal policies, as the synthetic nature of noise does not perfectly model human error or variability in performance.

Introduction of Distraction-Based Modeling: In response to the limitations of noise-based suboptimality, our work proposes the use of varying levels of distraction as a more ecologically valid method of inducing suboptimal demonstrations. By employing real-world distractions, such as concurrent tasks or environmental interruptions, we aim to capture a broader and more realistic spectrum of human performance degradation. This approach hypothesizes that modeling distractions can provide a clearer insight into the natural variability in human demonstrations, thereby leading to more accurate reward function recovery and policy optimization.

Comparative Studies: Our work is poised to compare the effectiveness of distraction-based suboptimality against traditional noise-based methods. Findings from our results suggest that distraction levels offer a more nuanced understanding of performance degradation, which is critical for refining LFD algorithms. This method's potential to recover more effective policies could be particularly beneficial in scenarios where robots must learn from non-expert human demonstrators in uncontrolled environments.

In conclusion, while noise-based methods have significantly advanced the field of LfD, the incorporation of distraction as a variable for inducing suboptimal demonstrations represents a promising frontier. Our project builds on the groundwork laid by prior studies, aiming to address their limitations and expand the practical applications of LfD.

3 Methods

3.1 Problem Formulation and Objective

Given a suboptimal demonstration which has a trajectory, τ_D, the objective is to attain the optimal policy, π^*. This policy is optimal with respect to an unknown true reward, R^*. Chen et al. [1] approach this problem by getting degrading trajectories: $\tau_1, \tau_2, ..., \tau_n$. Some noise η is used to generate the trajectories. The trend observed from the degraded performing trajectories is used to regress backwards to find R^*_{SSRR}, which is closer to R^*. One fundamental assumption is that the relation between an optimal demonstration, τ^* and D_S is caused by injection of η.

This works aims to challenge this assumption and empirically show that human suboptimality may follow other trends not captured by injection of η. To achieve this, varying levels of true human suboptimal demonstration are generated, $D_{L1}, D_{L2}, .., D_{Ln}$. The trend between these demonstrations is utilized to regress back to find a reward, R^*_{human}. If R^*_{human} is closer to $R*$ than R^*_{SSRR} ,then SSRR fails to capture the right relation between τ^* and τ_D. There exists a better way to generate degrading trajectories than injection with η. At least for this particular case.

3.2 Experimental Design

The study was conducted using four simulated environments from the OpenAI Gym toolkit: CartPole, Pendulum, Frozen Lake, and Car Racing Figure 1, 2, 3, and 4. These environments were selected to represent a range of tasks that require different types of control and strategic planning, making them ideal for testing the effects of distraction on performance. To introduce suboptimal demonstrations, we incorporated varying levels of distraction by requiring demonstrators to engage simultaneously in a secondary task, specifically a ”bubble pop” game, which requires visual attention and quick responses.

3.3 Observation and Reward function in simulation environments

In our algorithm, the environment Table 1 is assumed as a Markov Process. Thus, from observation space, reward can be explicitly calculated. For example, in the original Car Racing environment from Open Gym, the reward is -0.1 for every frame and +1000/N for every track tile visited. The positive reward is related to the direction of the road though there is no information from the observation. Thus, only by looking at the increment of reward, a player can know whether the car is heading in the right direction. This conflicts with the assumption. That's why we implemented a custom

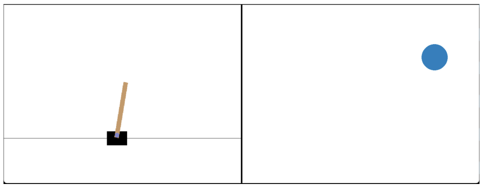


Figure 1: Cart Pole-Distraction Game

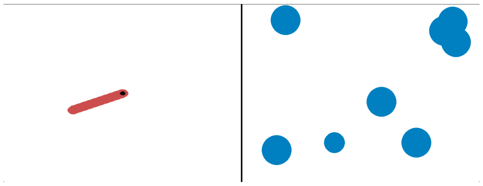


Figure 2: Pendulum-Distraction Game

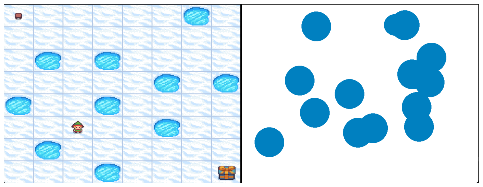


Figure 3: Frozen Lake-Distraction Game

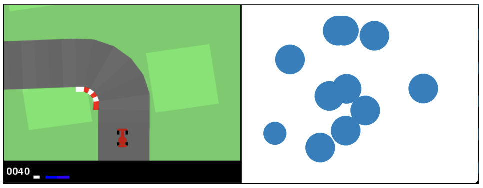


Figure 4: Car Racing-Distraction Game

OpenGym Environment	Action Space	Observation	Reward
Cart Pole	Left, Right Discrete(2)	Cart Position, Velocity Pole Angle, velocity	+1 if Pole Angle is less than +- 12 degree, otherwise 0
Pendulum	Torque **Continuous** between -2 and +2	$cos(theta)$, $sin(theta)$ Angular velocity	$r = -(theta^2 + 0.1 * \frac{dtheta}{dt} + 0.001 * torque^2)$
Frozen Lake	**Do nothing**, Left, Down, Right, Up Discrete(5)	agent's location in 8x8 grid world	+1 if agent reach to the goal, otherwise 0
Car Racing	**Do nothing**, Left, Right, Gas, and Brake Discrete(5)	**Correlation with gray** on screen and car **running status(bool)**	+1 if car is running on the track **both visited and un-visited**, otherwise 0

Table 1: Custom OpenGym environments

reward function for the Car Racing game. Also, the player requested to play distraction game at the same time. When the player move mouse to play distraction game, they may not control the agent in action space below. The action space or environment needs to provide maintain last action or 'do nothing' action. As a result, we added 'do nothing' on Frozen Lake.

3.4 Distraction Mechanism

The distraction task involved a simple game where circles (bubbles) appear at random locations on the screen at intervals and begin to shrink. Demonstrators were required to click these circles before they disappeared. The rate at which these circles appeared and the speed at which they shrank were adjusted to create different levels of distraction, categorized as Level 1 (low), Level 2 (medium), and Level 3 (high). The assumption was that higher levels of distraction would lead to more suboptimal demonstrations in the primary task.

3.5 Data Collection

Demonstrations were collected by recording the actions and states from each session in the gym environments. For each level of distraction, demonstrators performed multiple runs to ensure a robust dataset. The data collected included the actions taken by the demonstrators, the states of the environment at each step, and the resulting rewards.

3.6 Learning Algorithms

Figure 5 depicts our algorithm. We gathered demonstrations using varying levels of distraction. Adversarial Inverse Reinforcement Learning (AIRL) is then used on the demonstrations with the lowest distraction level to get a reward function. This reward function is applied to the other levels of demonstrations to generate the performance noise graphs. In our work, different from SSRR [1], noise is not added after demonstrations were done; instead, from the distraction level, demonstration itself contains certain level of noise. We will use an arbitrary scale to measure distraction. We perform curve fitting on the generated relationship between performance and distractions. Afterward, we apply reward regression to obtain a better reward function. Reinforcement learning is then used to find the optimal policy for this reward function. For reward regression, we adapted Categorical MLP Regressor for discrete action spaces from git hub in [1].

In more detail, our algorithm is divided into two primary learning algorithms.:

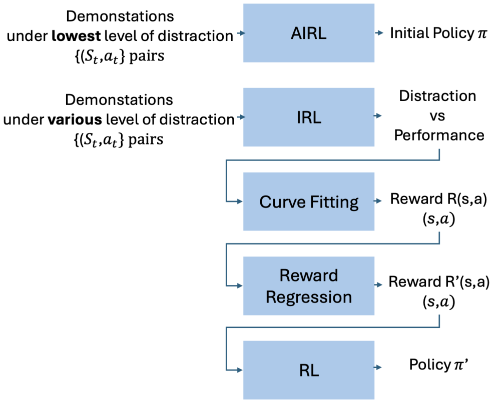


Figure 5: Proposed Algorithm

Behavior Cloning (BC): This supervised learning technique was used to directly mimic the demonstrator's actions. A simple multi-layer perceptron (MLP) model was trained on state-action pairs from the demonstrations.

Adversarial Inverse Reinforcement Learning (AIRL): This approach was used to infer the underlying reward function from the demonstrations at the lowest level of distraction. The inferred reward function was then used to generate policies for higher levels of distractions, aiming to understand how well the robot could adapt to suboptimal conditions.

3.7 Performance Evaluation

The effectiveness of the learned policies was evaluated by comparing the performance of the cart/pendulum/robot/car in the simulation environments without any distractions to the performance under simulated distractions. The metrics used for evaluation included the cumulative reward achieved by the policies and the deviation from the demonstrator's actions used in [3]. We are not providing results of cart-pole as they were not what we were expecting and we were getting very different and not usefull results.

3.8 Statistical Analysis

Statistical methods were employed to analyze the relationship between the levels of distraction and the performance of the demonstrations. Regression analysis was used to model this relationship, which helped in understanding the impact of distractions on the quality of demonstrations and the efficacy of the learning algorithms.

4 Experiments

Three main experiments were conducted. The first aims at analyzing whether different levels of suboptimality was introduced by increasing the difficulty of the distraction game. This relationship is observed by plotting the true reward achieved by each level of distraction. If the average cumulative reward for the demonstrations decreases with increase in distraction then varying levels of suboptimal demonstration has been created. The second experiment is to analyze the effectiveness of SSRR and D-REX to recover from the varying levels of suboptimality. Finally, our proposed method is evaluated against SSRR and D-REX.

4.1 Game choice

For Frozen Lake and Car Racing, a player can choose 'do nothing'. When the distraction level is increased, it is measured that the ratio of 'do notion' may be increased, accordingly in Figure 6. However, this trend may be an obstacle in learning from demonstration, because it means a seriously imbalanced class in the dataset. After we tried to learn MLP with Behavior Cloning, we found a few problems. First of all, reward is too sparse. Also, there is no time decaying in reward. Thus, even if the agent in the grid space stays as it is, there is no penalty on reward. Second, most of the actions are 'do nothing'. To make matters worse, the proportion of 'do nothing' was increased for the highest level of distraction. These things disturbed MLPs to learn from the demonstrations. As a result, for behavior, cloning of Frozen Lake, an agent in the environment was stuck at the start point and always chose to 'do nothing'. On the contrary, the custom Frozen lake in the Problem Set had the 'Monster' which acts like a time-decaying reward function as well as a facilitator to move for an agent.

However, for Car Racing, 'do nothing' wasn't a problem in behavior cloning. When we compare the ratio of 'do nothing' under three distraction levels in Figure 7, there was not much correlation between 'do nothing' and 'distraction level'. This can be explained by the rule of the environment. In the grid world of Frozen Lake, 'do nothing' doesn't change any environment. However, even the 'do nothing' action changes the environment in Car Racing because the velocity of the car is not changed to zero. Thus, when the road is straight, it is a natural thing to choose 'do nothing' action for a player in the environment.

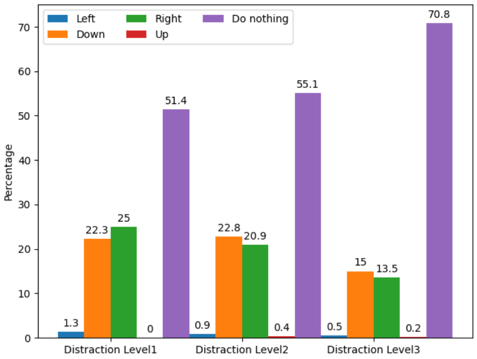


Figure 6: Distribution of Actions in Frozen Lake Demonstrations

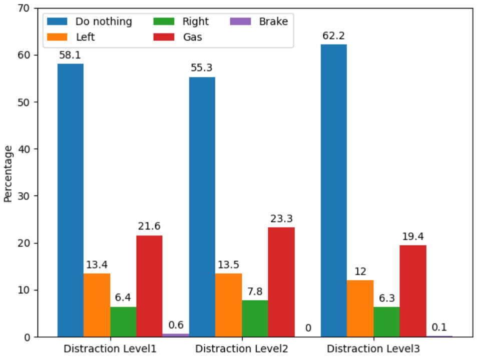


Figure 7: Distribution of Actions in Car Racing Demonstrations

	Level 2	Level 1	Level 3
D-REX	0.9727	0.9727	0.9393
SSRR	0.9523	0.941	0.829

Table 2: SSRR and D-REX performance for **three** level of distraction

5 Results

5.1 Varying levels of suboptimality

Figure 8 and figure 9 show the reward for demonstrations for varying levels of distraction game difficulty. The difficulty of distraction games does not seem to be enforcing performance degradation. Therefore, the speed of the game was used as another way to induce suboptimality. The step frequency in gym environments was increased to make it harder. Basically, the game would run at a faster rate requiring the user to respond faster. Figure 10 shows that this formulation enforces a degrading trend in demonstration quality.

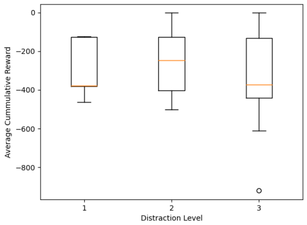


Figure 8: Pendulum for **three** levels of distraction

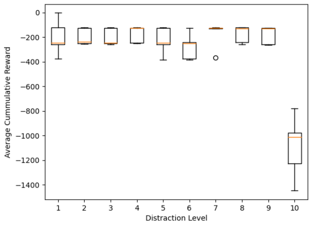


Figure 9: Pendulum for **ten** levels of distraction

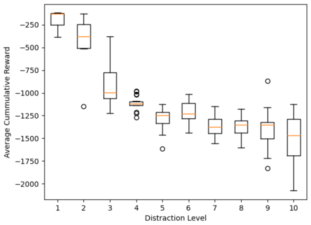


Figure 10: Pendulum for **ten** speed levels

5.2 SSRR and D-REX stress test

Demonstrations shown in figure 8 were used to test each of the algorithms. Since demos of level two outperformed level one, in table 2 the columns have been reordered to account for this. Surprisingly, D-REX outperforms SSRR. Possibly this is because D-REX does not make any assumption about the relationship between the demonstrations and just the ranking is taken into consideration. The assumption of ranking between τ^* and τ_D still holds true whether the demonstration is from a human or noise generated. D-REX shows a more generalized algorithm that could be preferable when the relation between τ_D and τ^* is unknown.

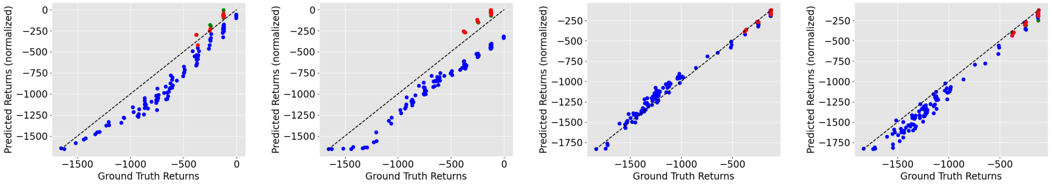
5.3 Noise Injection vs varying levels of suboptimal demonstration

Noise D-REX and Noise SSRR are methods that generate degrading trajectories by adding noise. Reaction D-REX and Reaction SSRR are the methods that use the demonstration gathered by varying speeds, figure 10. The term reaction is used as the suboptimality may be caused by the decreasing reaction time needed for better performance.

Using varying levels of suboptimality clearly outperforms noise-based methods. This empirically shows that modeling degrading performance by injection of η is not the best. The Table 3 Reported results are Mean (Standard Deviation) from five trials which are Learned Reward Correlation Coefficients and this metric is taken from [1]

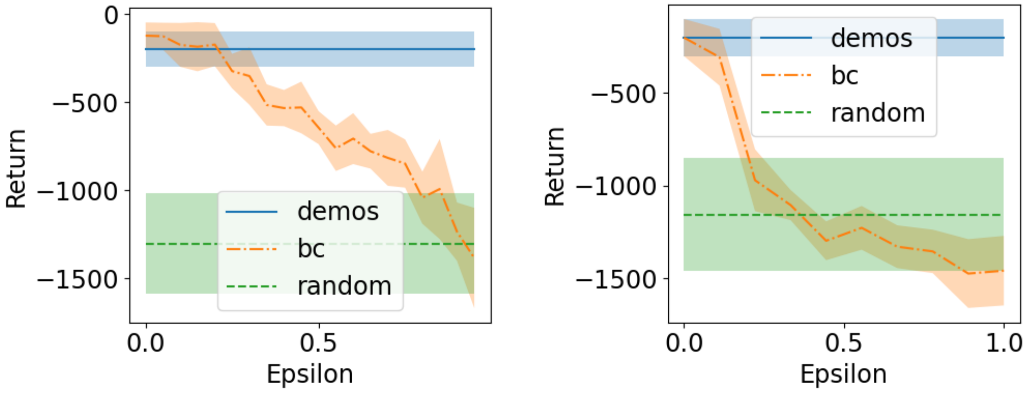
Method	Reward correlation R^2
Noise D-REX	0.94224
Noise SSRR	0.93456
Reaction D-REX	0.99722
Reaction SSRR	0.99398

Table 3: Performance of SSRR and D-REX for Noise and Reaction modeling



(a) Noise D-REX (b) Noise SSRR (c) Reaction D-REX (d) Reaction SSRR

Figure 11: Plot of ground truth reward vs reward obtained by the algorithm



(a) Noise D-REX (b) Noise SSRR

Figure 12: Plot of ground truth reward vs reward obtained by the algorithm

6 Conclusion

The results affirm the hypothesis that integrating realistic distractions into training scenarios can enhance the robustness of derived policies. Notably, policies trained under moderate distractions generalized better to un-distracted environments than those trained under high distractions or no distractions.

This research underlines the potential of distraction-based training in enhancing the adaptability of learning algorithms. It challenges existing assumptions in IRL that optimal performance in training leads to optimal real-world performance.

This study demonstrated that distraction-based training could lead to the development of more adaptive and resilient robotic systems compared to traditional noise-based training methods.
Future work includes getting on more demonstrations through various levels of expertise and analyzing the trend for better understanding. Look for potential new methods for comparison and improve the methods for demonstration.

References

[1] L. Chen, R. Paleja, and M. Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. *Conference on robot learning(CoRL)*, pages 1262–1277, 2021.

[2] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard. Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 2020.

[3] J. S. S.-H. S. M. Gombolay, R. Jensen and J. Shah. Apprenticeship scheduling: Learning to schedule from human experts. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[4] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. *International Conference on Machine Learning (ICML)*, 2004.

[5] B. D. Ziebart, A. Maas, J. A. Bagnell, and D. A. K. Maximum entropy inverse reinforcement learning. *National Conference on Artificial intelligence (AAAI)*, 2008.

[6] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

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

[8] C. Wirth, R. Akrour, G. Neumann, and J. Furnkranz. A survey of preference-based reinforcement learning ¨ methods. *Journal of Machine Learning Researchs*, page 18:1–46, 2017.