Learning to Weight Imperfect Demonstrations

Yunke Wang [1] **Chang Xu** [2] **Bo Du** [1] **Honglak Lee** [3 4]

Abstract

This paper investigates how to weight imperfect expert demonstrations for generative adversarial imitation learning (GAIL). The agent is expected to perform behaviors demonstrated by experts. But in many applications, experts could also make mistakes and their demonstrations would mislead or slow the learning process of the agent. Recently, existing methods for imitation learning from imperfect demonstrations mostly focus on using the preference or confidence scores to distinguish imperfect demonstrations. However, these auxiliary information needs to be collected with the help of an oracle, which is usually hard and expensive to afford in practice. In contrast, this paper proposes a method of learning to weight imperfect demonstrations in GAIL without imposing extensive prior information. We provide a rigorous mathematical analysis, presenting that the weights of demonstrations can be exactly determined by combining the discriminator and agent policy in GAIL. Theoretical analysis suggests that with the estimated weights the agent can learn a better policy beyond those plain expert demonstrations. Experiments in the Mujoco and Atari environments demonstrate that the proposed algorithm outperforms baseline methods in handling imperfect expert demonstrations.

1. Introduction

Imitation learning (IL) (Abbeel & Ng, 2004; Argall et al., 2009; Hussein et al., 2017), which aims to let the agent imitate the behavior of the human demonstrations without any access to reward signal, has achieved great success in many sequential decision making problems (Stadie et al., 2017; Ermon et al., 2015; Finn et al., 2016). Compared to complex reward engineering (Ng et al., 1999; Amodei et al., 2016) in reinforcement learning (RL) (Sutton & Barto, 2018), IL provides a much easier way to infer a reward function from the collected demonstrations directly, so that the agent can effectively improve the demonstrated behavior.

Generative adversarial imitation learning (GAIL) (Ho & Ermon, 2016) is one of the state-of-the-art imitation learning methods. Within the framework of generative adversarial network (GAN) (Goodfellow et al., 2014), GAIL regards imitation learning as a distribution matching problem between the state-action distribution of expert policy and that of agent's policy. After GAIL, a few variants have been developed to further improve the performance of plain GAIL from different aspects. InfoGAIL (Li et al., 2017) learns a policy with a latent variable, and the latent variable is believed to have interpretable representations of complex demonstrations. This allows InfoGAIL to reproduce a variety of behaviors within demonstrations. Since it is hard to recover the reward function with GAIL, Adversarial IRL (Fu et al., 2017) separates a reward function from the discriminator, and thus the reward function can be directly learned during training. AIRL is also proven to be effective and robust in large and high-dimension dynamics. WAIL (Xiao et al., 2019) follows the idea of WGAN (Gulrajani et al., 2017) by replacing the Jensen-Shannon (JS) divergence objective function with a Wasserstein distance GAN-based objective function, so that WAIL can directly learn a reward function in a more stable way.

Though GAIL and its variants have achieved impressive experimental results, their underlying assumption that the expert demonstrations are sampled from an optimal policy cannot always hold in practice. In real-world tasks, it is usually difficult to collect plenty of expert demonstrations from the optimal policy. Even a human expert cannot always make optimal choices due to limited energy and the presence of distractions. For example, an agent learns how to play football through imitation learning, where professional football players can be qualified experts. However, since professional football players may still make mistakes during the match, the resulting imperfect expert demonstrations could thus mislead the procedure of imitation learning

[1]National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, School of Computer Science and Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China [2]School of Computer Science, Faculty of Engineering, The University of Sydney, Australia [3]EECS Department, University of Michingan, USA [4]LG AI Research, South Korea. Correspondence to: Chang Xu <c.xu@sydney.edu.au>, Bo Du <dubo@whu.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

(Ratliff et al., 2006).

There are some attempts to address the imperfect demonstrations issue. In 2IWIL (Wu et al., 2019) and IC-GAIL (Wu et al., 2019), a fraction of demonstrations labeled with confidence score need to be provided first in training. Preference-based inverse reinforcement learning methods such as T-REX (Brown et al., 2019) need ranked trajectories to learn a relevant reward function and then conduct RL with the new learned reward. Multi-modal imitation learning method InfoGAIL can be used to recover all the demonstrators within demonstrations, however a prior about the number of demonstrators is needed first and we also cannot find the best demonstrator before we evaluate all the modes. Overall, most of these existing solutions to address the imperfect expert demonstrations problem heavily rely on the availability of all kinds of prior information (e.g., labeled data by an oracle or preference rankings of demonstrations) or some strong assumptions on expert demonstrations, which cannot be naturally satisfied in practice.

In this paper, we propose a new approach based on generative adversarial imitation learning framework to handle imperfect expert demonstrations without any prior. More specifically, our method can automatically predict weight for each expert demonstration to assess its quality and importance for agent learning. We conduct mathematical analysis to show that the weight can be well estimated by the discriminator and agent policy in GAIL. In the training procedure, weight estimation and agent policy learning interact and are thus optimized as a whole, which improves the performance of imitation learning. Compared to existing solutions, our method finds the connection between weight estimation and plain GAIL, thus we can predict weight without prior information. Theoretical analysis suggests that an improved expert policy can be produced to benefit the learning of the agent. Experiment results on Mujoco (Todorov et al., 2012) and Atari demonstrate that the proposed method can better exploit imperfect demonstrations than comparison methods.

2. Related Work

Imitation learning can be roughly divided into three categories. Behavioral Cloning (BC) (Bain & Sammut, 1995) regards state and action of each expert demonstration as input and output, then learns a policy model in a supervised learning fashion. BC is easy to perform but it ignores the association between states (Ross et al., 2011) and often requires vast amounts of data for training a good policy. The idea of inverse reinforcement learning (IRL) (Abbeel & Ng, 2004; Ziebart et al., 2008) is to reconstruct the reward function by assuming the given expert demonstrations are sampled from the optimal policy. General RL methods can then be applied to update the agent's policy after the reconstructions of the reward function. GAIL is one of the state-of-the-art methods, which connects IRL methods to GAN framework.

Imitation learning is an interdisciplinary field of research that can be widely applied to a variety of tasks in the real world such as robot control (Englert et al., 2013) and autonomous driving (Giusti et al., 2015; Codevilla et al., 2018) with good performance. However, good performance of the agent in imitation learning often requires demonstrations which are of high quality, while collecting high-quality demonstrations can be a difficult task. So imitation learning from imperfect demonstrations becomes an important issue in IL. Several relevant researches to address this issue are discussed below, however most of these methods require auxiliary information, which constrains them to be applied to more universal settings.

2.1. Imperfect Demonstrations issue

IRL methods based on preference learning (Sugiyama et al., 2012; Wirth et al., 2016; Christiano et al., 2017) can be used to deal with imperfect demonstrations. Suppose a set of ranked trajectories is given for training, T-REX (Brown et al., 2019) aims to recover a reward function that can well fit the ranking of trajectories. That is to say, a better trajectory should relate to a higher estimated cumulative reward. Considering this rank as auxiliary information that may not be provided in ordinary IL tasks, D-REX (Brown et al., 2020) is proposed to automatically get this rank. D-REX needs a base policy trained by BC first and then derives noisy policy to generate ranked trajectories. The base policy is learned over imperfect demonstrations thus directly influences the performance of D-REX.

Suppose a fraction of demonstrations are labeled with confidence to indicate whether they belong to the optimal demonstrations, 2IWIL (Wu et al., 2019) trains a semi-supervised classifier to predict confidence score for unlabeled demonstrations and then a weighted GAIL framework is used to train the agent. To avoid accumulated error in two steps, IC-GAIL (Wu et al., 2019) is proposed to train in an end-to-end fashion compared to 2IWIL. With a fraction of labeled demonstrations, SSIRL (Valko et al., 2013) uses semi-supervised support vector machines to separate good or bad demonstrations when learning a policy, thus improving the performance of the agent. Consider weighting trajectories, RBIRL (Zheng et al., 2014) seeks to infer weights on trajectories level to perform better imitation learning.

If we consider demonstrations are from multiple demonstrators, we may have multi-modal imitation learning problems. VILD (Tangkaratt et al., 2019) models the distribution of multi-modal demonstrations with a rigorous assumption that each demonstrator is shaped by adding different Gaussian noise to the optimal policy. Thus it may not work well under more universal settings. InfoGAIL (Li et al.,

2017) has shown to be able to recover all the modals inside demonstrations, however we need to go through all the modals before we find the best one. Compared with mode-covering in GAIL, AIRL (Fu et al., 2017) could lead to a mode-seeking behavior by minimizing reverse KL divergence, which might be helpful to seek the best mode. However, the sought mode is arbitrary, which depends on the initialization and mode mixture (Ke et al., 2019). It is unlikely to guarantee the sought mode is the good one.

3. Preliminaries

In this section, we briefly introduce the framework of reinforcement learning and the generative adversarial imitation learning method.

3.1. Reinforcement Learning

Reinforcement learning aims to learn an optimal policy for the agent while the agent explores the environment and gets feedback to adjust its policy. The framework of reinforcement learning is generally based on the Markov Decision Process (MDP) (Puterman, 2014). An MDP consists of five elements $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma \rangle$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the transition probability distribution, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1]$ is the discounting factor for future rewards. The return in the MDP is calculated as the discounted sum of rewards obtained by the agent over all episodes,

$$\eta(\pi) = \mathbb{E}_{\pi}\big[\sum_{t=0}^{T} \gamma^t r(s_t, a_t)\big] \tag{1}$$

where $s_t \in \mathcal{S}$ denotes a state vector, $a_t \in \mathcal{A}$ is an action vector and $r(s_t, a_t)$ stands for the one-step reward when the agent is at the state s_t and takes the action a_t. We define $\tau = \{(s_0, a_0), (s_1, a_1), \cdots, (s_T, a_T)\}$ as a trajectory of the states and actions. The goal of RL is thus to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that can maximize the expected return over all episodes. For any policy π, there is a one-to-one correspondence between the policy and its occupancy measure $\rho_\pi : S \times A \to R$. We also define state-value function as $V_\pi(s_t) = \mathbb{E}_{\pi|s_t}\big[\sum_{l=0}^{T} \gamma^l r(s_{t+l}, a_{t+l})\big]$, action-value function as $Q_\pi(s_t, a_t) = \mathbb{E}_{\pi|s_t,a_t}\big[\sum_{l=0}^{T} \gamma^l r(s_{t+l}, a_{t+l})\big]$ and advantage function as $A_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t)$.

3.2. Generative Adversarial Imitation Learning

Imitation learning aims to learn an optimal policy π_θ for the agent based on the trajectories $\{(s_i, a_i)\}_{i=1}^{T}$ sampled from the expert policy π_E. The general framework of Generative Adversarial Networks (GANs) had been applied in the imitation learning problem, which results in the Generative Adversarial Imitation Learning (GAIL) algorithm.

In GAIL, the classical imitation learning problem is treated as an occupancy measure matching between the expert policy and the agent policy via Jensen-Shannon Divergence. A discriminator D_ψ is introduced to distinguish expert transitions from agent transitions, while the agent is to "fool" the discriminator into taking agent transitions as those expert transitions. Formally, the objective function of GAIL is written as

$$\min_\theta \max_\psi \ \mathbb{E}_{(s,a)\sim\rho_{\pi_\theta}}[\log D_\psi(s, a)] + \mathbb{E}_{(s,a)\sim\rho_{\pi_E}}[\log(1 - D_\psi(s, a))], \tag{2}$$

where ρ_{π_θ} and ρ_{π_E} denote the distributions of the agent policy π_θ and the expert policy p_E respectively. The agent is trained to minimize $\mathbb{E}_{(s,a)\sim\rho_{\pi_\theta}}[\log D_\psi(s, a)]$, and the output of the discriminator $-\log D_\psi(s, a)$ can be taken as the reward. Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) or other policy gradient RL methods (Lillicrap et al., 2015; Schulman et al., 2017) can be used to update agent policy π_θ.

4. Methodology

GAIL encourages the agent to imitate expert demonstrations. But if the collected expert demonstrations are imperfect, the resulting learned agent policy could be seriously influenced. We suppose an imperfect demonstration is a mixture sampled from an optimal policy and a non-optimal policy, with different proportions of "optimal", imperfect demonstrations should have different quality. Also, these demonstrations with different qualities may have different contributions and impacts on agent learning. To solve this, a common way is to assign a weight w to each demonstration to address its importance and contribution to learn an agent.

Inspired by (Wu et al., 2019), which presents a weighted framework for GAIL with confidence scores labeled by an oracle, we write a more general weighted GAIL objective function as follows

$$\min_\theta \max_\psi \ \mathbb{E}_{(s,a)\sim\rho_{\pi_\theta}}[\log D_\psi(s, a)] + \mathbb{E}_{(s,a)\sim\rho_{\pi}}[w(s, a)\log(1 - D_\psi(s, a))], \tag{3}$$

where $w(s, a)$ denotes the weight of state-action pair (s, a), π_θ is the agent policy and π can be regarded as a mixture of policies with different qualities. It is easy to collect state-action pairs from π, however, there is no access to get the weight for each state-action pair directly. Before we study how to approximate the weight, we first proceed to give an in-depth discussion on the advantages of imitation learning with weights and what an ideal weight would look like.

4.1. Learning from A New Policy

In weighted GAIL (Eq. (3)), we assign a weight $w(s, a)$ to the output $\log(1 - D_\psi(s, a))$ of each state-action pair sam-

pled from the expert. But how the weighted GAIL enables us to learn from the imperfect expert demonstration? We begin with the introduction of f-divergence, which is widely used to measure similarity between probability distributions. A general form of f-divergence can be written as,

$$D_f(p,q) = \sum_x q(x) f(\frac{p(x)}{q(x)}) \tag{4}$$

where p and q are the distributions of the variable x. Different f functions can recover different divergences, i.e. Jensen-Shannon (JS) divergence $f_{JS}(u) = -(u+1)\log\frac{u+1}{2} + u\log u$ and Kullback–Leibler (KL) divergence $f_{KL}(u) = u\log u$. The convex conjugate of $f(x)$ is defined as $f^*(y) = \sup_{x\in dom_f}(\langle y, x\rangle - f(x))$.

By defining $T(s,a) = \log(D_\psi(s,a))$, we can apply the convex conjugate of JS divergence f^*_{JS} , and rewrite weighted GAIL in Eq. (3) as

$$\min_\theta \sup_{T\in\mathcal{T}} \Big(\mathbb{E}_{(s,a)\sim\rho_{\pi_\theta}}[T(s,a)] - \mathbb{E}_{(s,a)\sim\rho_\pi}[w(s,a) f^*_{JS}(T(s,a))]\Big). \tag{5}$$

By absorbing the weight $w(s,a)$ into ρ_π, we can define the objective of weighted GAIL as $J\big(T(s,a)\big)$

$$J\big(T(s,a)\big) = \sup_{T\in\mathcal{T}} \Big(\mathbb{E}_{(s,a)\sim\rho_{\pi_\theta}}[T(s,a)] - \mathbb{E}_{(s,a)\sim\rho_{\widetilde{\pi}}}[f^*_{JS}(T(s,a))]\Big), \tag{6}$$

where $\rho_{\widetilde{\pi}}(s,a) = w(s,a)\rho_\pi(s,a)$ stands for a new policy based on π. Due to the Jensen's inequality and the restriction of class function $T \in \mathcal{T}$, we obtain the upper bound of $J\big(T(s,a)\big)$ as

$$\sum_{s,a} \rho_{\widetilde{\pi}}(s,a) \sup_{T(s,a)\in dom_{f^*}} \Big(T(s,a)\frac{\rho_{\pi_\theta}(s,a)}{\rho_{\widetilde{\pi}}(s,a)} - f^*_{JS}(T(s,a))\Big). \tag{7}$$

According to the definition of the convex conjugate of JS divergence, we have the definition of $f_{JS}\big((\rho_{\pi_\theta}(s,a))/(\rho_{\widetilde{\pi}}(s,a))\big)$ as

$$\sup_{T(s,a)\in dom_{f^*}} \big(T(s,a)\frac{\rho_{\pi_\theta}(s,a)}{\rho_{\widetilde{\pi}}(s,a)} - f^*_{JS}(T(s,a))\big), \tag{8}$$

based on which, Eq. (7) can be rewritten as

$$\begin{aligned} J\big(T(s,a)\big) \le & \sum_{s,a} \rho_{\widetilde{\pi}}(s,a) f_{JS}(\frac{\rho_{\pi_\theta}(s,a)}{\rho_{\widetilde{\pi}}(s,a)}) \\ = & \; D_{JS}(\rho_{\pi_\theta}, \rho_{\widetilde{\pi}}). \end{aligned} \tag{9}$$

Imitation learning can be also considered as an occupancy measure matching problem (Ke et al., 2019). JS divergence here provides an effective approach to measure the similarity between expert policy and the learned agent policy. Minimizing JS divergence $D_{JS}(\rho_{\pi_\theta}, \rho_{\widetilde{\pi}})$ can thus be regarded as a process to imitate a new policy $\rho_{\widetilde{\pi}}$ for the agent.

4.2. Analysis on The Weight

We have known that learning from expert policy π with weighted GAIL is equivalent to imitate a new policy $\widetilde{\pi}$, with $\rho_{\widetilde{\pi}}(s,a) = w(s,a)\rho_\pi(s,a)$ satisfied. In practice, the expert policy π could generate an imperfect demonstration that impacts the agent policy learning. How to formulate the new policy $\widetilde{\pi}$ to guarantee its advantage over the plain expert policy is therefore important.

Recall that in Eq. (1), $\eta(\pi)$ has been defined as the expected reward over all episodes of policy π. $\eta(\widetilde{\pi}) - \eta(\pi)$ can be naturally used to measure the performance gap between two policies $\widetilde{\pi}$ and π. Since it is hard to sample from state visiting distribution of $\widetilde{\pi}$, we can sample from π approximately (Schulman et al., 2015). Formally, we approximate $\eta(\widetilde{\pi}) - \eta(\pi)$ by $L^{d_\pi}(\widetilde{\pi})$, and have

$$L^{d_\pi}(\widetilde{\pi}) = \sum_s d_\pi(s) \sum_a \widetilde{\pi}(a|s) A_\pi(s,a), \tag{10}$$

where $d_\pi(s) = \mathbb{E}_\pi\big[\sum_{t=0}^T \gamma^t \mathbf{1}(s_t = s)\big]$ is the state visiting distribution of policy π and $\mathbf{1}(\cdot)$ is an indicator function.

To derive a new policy $\widetilde{\pi}$ that performs better than π, we need to maximize $L^{d_\pi}(\widetilde{\pi})$. Meanwhile, as the new policy $\widetilde{\pi}$ stems from the original policy π, their probability distribution shall not be too far from each other. We therefore obtain an f-divergence constraint policy improvement problem in the following theorem to derive a new policy $\widetilde{\pi}$.

Theorem 1. *Consider an f-divergence constrained policy optimization problem as,*

$$\begin{aligned} \max_{\widetilde{\pi}} \quad & L^{d_\pi}(\widetilde{\pi}) - \beta D_f(\rho_{\widetilde{\pi}} || \rho_\pi) \\ s.t. \quad & \sum_a \widetilde{\pi}(a|s) = 1, \widetilde{\pi}(a|s) \in [0,1] \end{aligned} \tag{11}$$

where β is a hyper-parameter to balance the influence of these two terms. We have,

$$\widetilde{\pi}(a|s) = \pi(a|s) f'_*\big((1/\beta)(A_\pi(s,a) + C(s))\big) \tag{12}$$

Theorem 1 provides a solution of f-divergence constraint policy optimization problem in general form, the proof is provided in supplementary material. Since the general f is not intuitive, we consider a special case of f-divergence instead. When using KL-divergence, $f'_*(u) = e^u$, Eq. (12) can be rewritten as follows

$$\widetilde{\pi}(a|s) = \pi(a|s)\exp((1/\beta)(A_\pi(s,a) + C(s))). \tag{13}$$

So far we have a new policy $\widetilde{\pi}$ derived from Theorem 1. Intuitively, Eq. (13) shows that the state-action pair which has higher advantage in π is more likely to occur in $\widetilde{\pi}$. In the following theorem we will further discuss the new policy $\widetilde{\pi}$ is indeed better than π.

Theorem 2. *Given two policies $\widetilde{\pi}$ and π which satisfies*

$$\widetilde{\pi}(a|s) = \pi(a|s)\exp((1/\beta)(A_\pi(s,a) + C(s))) \quad (14)$$

where β is a hyper-parmeter and $C(s)$ is a function of state s. We can conclude that $\widetilde{\pi}$ is generally better than π, that is,

$$V_{\widetilde{\pi}}(s) \geq V_\pi(s), \forall s \in \mathcal{S}. \quad (15)$$

Theorem 2 suggests that $\widetilde{\pi}$ is indeed better than π, and the proof is provided in supplementary material. To recover the occupancy measure, Eq. (13) can be rewritten as $\rho_{\widetilde{\pi}}(s,a) = \rho_\pi(s,a)\exp((1/\beta)A_\pi(s,a))$, where $d_{\widetilde{\pi}}(s) \propto d_\pi(s)\exp(C(s))$. Recall that we have proved that weighted GAIL is equivalent to imitating policy $\widetilde{\pi}$ which satisfies $\rho_{\widetilde{\pi}}(s,a) = w(s,a)\rho_\pi(s,a)$. Thus, we can conclude that $\widetilde{\pi}$ is indeed better than π if $w = \exp((1/\beta)A_\pi(s,a))$ is satisfied. Also, we can easily find $w \propto A_\pi$, which means the state-action pairs which have a higher advantage in MDP should also have higher weight w. In other words, a better state-action pair should have a higher weight.

4.3. Learning to Weight

Given the connection between weight w and advantage $A_\pi(s,a)$, we next proceed to estimate the advantage within GAIL framework and suggest that the weight can be well approximated with discriminator D_ψ and agent policy π_θ.

The optimization of Eq. (3) is a minimax optimization problem and we define the solution of the inner maximization problem as the optimal discriminator D^*_ψ during the intermediate stage of the whole optimization. Therefore, D^*_ψ should be able to well distinguish the demonstrations between π_θ and $\widetilde{\pi}$. Instead of using occupancy measure, we consider demonstrations sampled from action distribution in the implementation. The algorithm involves a warm-up step (the weight is set to 1 for each demonstration), after which $d_\pi(s)$ and $d_{\widetilde{\pi}}(s)$ could have part of match. Given $\pi(a|s) = \rho_\pi(s,a)/d_\pi(s)$, we approximate D^*_ψ as,

$$D^*_\psi(s,a) = \frac{\pi_\theta(a|s)}{\pi_\theta(a|s) + w(s,a)\pi(a|s)} \quad (16)$$

$$= \frac{\pi_\theta(a|s)}{\pi_\theta(a|s) + w(s,a)\exp(A_\pi(s,a))}. \quad (17)$$

The second equation can be satisfied since the advantage function $A_\pi(s,a)$ has been suggested to be recovered by $\log[\pi(a|s)]$ (Fu et al., 2017). Recall that $w = \exp((1/\beta)A_\pi(s,a))$ is satisfied, so the advantage can be also written as $A_\pi(s,a) = \log w(s,a)^\beta$. Combined with Eq. (16), we can further write D^*_ψ as,

$$D^*_\psi(s,a) = \frac{\pi_\theta(a|s)}{\pi_\theta(a|s) + w(s,a)^{(\beta+1)}} \quad (18)$$

Thus the weight w can be finally written as

$$w(s,a) = [(1/D^*_\psi(s,a) - 1)\pi_\theta(a|s)]^{\frac{1}{\beta+1}}, \quad (19)$$

where $\pi_\theta(a|s)$ denotes the probability of agent policy to take action a at state s. To calculate w, the optimal discriminator D^*_ψ and the agent policy π_θ are needed. However the GAIL training should carefully balance the policy and the discriminator, we may not have access to D^*_ψ. As (Peng et al., 2018) suggested, the discriminator has shown to easily overwhelm the policy during the GAIL training procedure, especially in the early training stage. This suggests D_ψ under this condition is discriminative, and it can therefore act as an approximation of D^*_ψ. Thus, we conduct early stop to weight estimation during the training to satisfy this approximation of D^*_ψ.

While replacing advantage fucntion with $\log \pi(a|s)$ in Eq. (13), the new formed policy $\widetilde{\pi}$ can be actually regarded as a lower-temperature version of π. This suggests that importance weighting in weighted GAIL could lead agent to imitate a more discriminative distribution beyond ρ_π. The resulting 'less randomness' implies that one state-action pair of a lower probability will be even more unlikely.

According to Eq. (19), the larger $\pi_\theta(a|s)$ and a lower $D_\psi(s,a)$ will lead to a higher weight $w(s,a)$. It is natural to see that $w \propto (1/D_\psi)$ is satisfied since expert demonstrations should be always assigned to a lower D_ψ. As for π_θ, it would be larger for those pairs of smaller $D_\psi(s,a)$ to minimize $\mathbb{E}_{\rho_{\pi_\theta}}[D_\psi(s,a)]$, as they look more like the expert demonstrations. That is to say, the agent tends to learn more from its past good experience (justified by the smaller $D_\psi(s,a)$), which is consistent with the findings of self-imitation learning (Oh et al., 2018). In practice, optimal demonstrations could have regular patterns and strong clustering effects, while mistakes in sub-optimal demonstrations often differ from one another. If most demonstrations are optimal, we can easily learn D_ψ and assign smaller weights to those few sub-optimal demonstrations, which then leads to a curriculum learning (Bengio et al., 2009) or self-paced learning (Kumar et al., 2010). If we are overwhelmed with sub-optimal demonstrations, their non-significant patterns might not always win them a smaller $D_\psi(s,a)$ or larger $w(s,a)$ than those few optimal demonstrations.

Dynamic Importance Weighting. By now we have found the connection between weight estimation and plain GAIL problem, then we put weight estimation for imperfect demonstrations and the agent policy training into an interaction framework and optimize them as a whole. More specifically, in each iteration, the weight estimation is conducted first to predict weight for each demonstration with Eq. (19). Then weighted GAIL (Eq. (3)) proceeds to train the agent policy with weighted demonstrations. These two procedures interact alternately, the agent policy gradually improves its performance while the weight also fixes itself dynamically and both of them will reach convergence within this alternating optimization problem.

This dynamic importance weighting procedure is highly related to some previous meta-learning works, which alternates between weight estimation and weighted classification. As discussed in (Fang et al., 2020) and (Ren et al., 2018), importance weighting is an effective way to handle distribution shift problem within training and validation set. Similarly, matching a new policy $\widetilde{\pi}$ beyond π with weight w can be also viewed as a procedure of solving the distribution shift problem in imitation learning by importance weighting. The difference is that the access to validation set in (Ren et al., 2018) and (Fang et al., 2020) makes it easier to estimate weight, however in weighted GAIL the target policy is not available and we need to design its form first.

Connection with 2IWIL. Our method shares some similarities with 2IWIL (Wu et al., 2019), which also conducts a weighted GAIL framework. We address the key difference between 2IWIL and our method from the following two aspects. First, a fraction of demonstrations labeled with confidence should be given first in 2IWIL, which means 2IWIL needs auxiliary information to perform imitation learning tasks. By contrast, our weighted GAIL finds the connection between weight estimation and plain GAIL, thus we can predict weight for demonstration in an unsupervised way. Second, the weight definitions are different. Based on importance sampling, 2IWIL defines the weight as $w(s,a) = c(s,a)/\alpha$, where $c(s,a)$ denotes the probability of the demonstration (s,a) to be optimal and α denotes the mean of confidence score. By contrast, the weight in Eq. (19) is designed by the motivation of learning a better policy beyond plain expert policy with theoretical supports.

5. Experiment

In this section, we conduct experiments on both Mujoco and Atari to evaluate the performance of weighted GAIL (WGAIL). We are going to figure out two questions: (1) Can WGAIL show significant improvement compared to original GAIL from different aspects? (2) What is the learned weight like? We provide detailed results and discussions below.

5.1. Experiment with Mujoco

We first conduct experiments on four continuous control tasks in the Mujoco simulator (Todorov et al., 2012): Ant-v2, Hopper-v2, Walker2d-v2, and HalfCheetah-v2. The training agent in Mujoco aims to run as far as possible, so we use this indicator to show the performance of the learned agent. To reduce uncertainty, we evaluate the new-learned agent's policy three times in the evaluation. Also, we conduct our experiment with five different random seeds.

Data Collection. Our setting is to handle imperfect demonstrations, which can be regarded as a mixture sampled from different sub-optimal policies. So first we use TRPO to train an optimal policy and save 3 checkpoints during training, then the "Stage 1" demonstrations are formed by equal demonstrations from checkpoint 1 and checkpoint 2 while the "Stage 2" demonstrations are equal sampled from checkpoint 1 and checkpoint 3. Notice that checkpoint 3 is the optimal policy. Therefore, "Stage 1" and "Stage 2" demonstrations can represent different levels of demonstrations and the latter is obviously better.

Compared Methods. GAIL, BC, D-REX, 2IWIL and T-REX are compared in our experiment. The formal three methods can be conducted in standard IL setting while the latter two methods need auxiliary information. GAIL, BC, and D-REX share the same setting as weighted GAIL. For T-REX, it needs the cumulative MDP reward of trajectories to rank the trajectories normally, however, the reward is not available in our setting. Thus we give a time prior to T-REX and the trajectories are ranked by time-index checkpoints. For example, trajectories sampled from checkpoint 3 are regarded to be better than trajectories sampled from checkpoint 2. Semi-supervised method 2IWIL needs confidence score of demonstrations. To satisfy this need, we use the quality (normalized reward) of a checkpoint as a coarse estimation of the confidence score for demonstrations sampled from this checkpoint.

5.1.1. Performance

The result in Table 1 shows the great performance of WGAIL. In most cases, WGAIL can outperform other compared methods. In each task, the result of WGAIL in "Stage 2" is always better than "Stage 1", which suggests that the performance of WGAIL may improve with the increasing quality of demonstrations. We also provide an intuitive model optimization trajectories map via a 2D weight-space

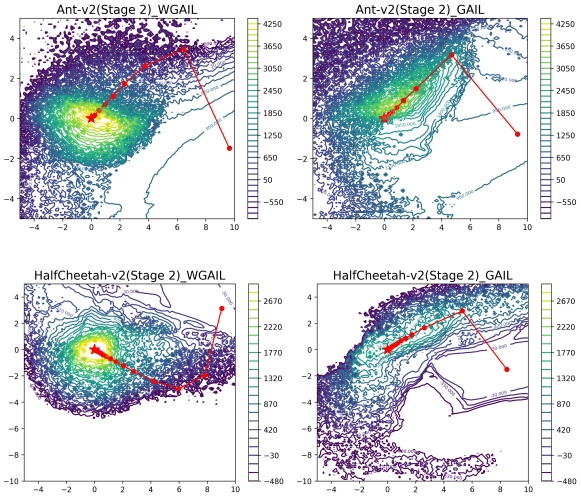

Figure 1. The optimization trajectories of policy models and reward surface in WGAIL and GAIL. The dots denote the location of the policy checkpoints every 500 epochs.

Method	Ant-v2		HalfCheetah-v2		Walker2d-v2		Hopper-v2	
	Stage 1	Stage 2	Stage 1	Stage 2	Stage 1	Stage 2	Stage 1	Stage 2
WGAIL	**111.81**	**182.00**	**120.66**	**190.17**	5.29	**18.11**	**12.08**	**14.54**
GAIL (Ho & Ermon, 2016)	84.50	113.58	102.21	85.33	6.66*	10.80	10.06	11.81
BC (Bain & Sammut, 1995)	91.49	135.75	99.2	118.15	**7.61***	12.22	0.53	0.95
D-REX (Brown et al., 2020)	48.63	63.43	28.73	84.57	3.10	9.61	2.59	2.17
T-REX (Brown et al., 2019)	65.08	9.25	95.57	32.70	-0.12	-0.46	5.70	1.14
2IWIL (Wu et al., 2019)	89.03	130.14	101.43	85.94	7.63*	11.66	11.02	17.54*
Expert (TRPO)	177.93		195.79		24.71		18.63	

Table 1. Performance of the learned agent, measured by the final x-position of the agent's body. The final x-position of WGAIL, GAIL and 2IWIL is calculated by the average of the last 50 evaluations.

slice in Figure 1, from which we can observe that WGAIL's final model is always located at high reward zone while the model of GAIL is not. The optimization trajectories are plotted along the contours of the average reward surface for this slice. To plot this figure, we follow the instruction of (Li et al., 2018).

We use fewer demonstrations to train the agent in a simple task, which might be the reason why BC does not perform well in Hopper-v2. It shows BC will need more demonstrations to reach a great performance compared to GAIL. We also notice that D-REX and T-REX, two preference-based IRL methods do not perform well as expected. D-REX needs a base policy trained by BC first to generate ranked trajectories. If given demonstrations are imperfect, the performance of base policy can be influenced, which might make it hard to generate the precise ranking. As for T-REX, we think the time-based preference is quite coarse-grain and two demonstrators are not enough for inferring a good reward function, sometimes it may overfit. 2IWIL performs generally better than plain GAIL. The added confidence score gives more information and has a positive impact on agent learning. However, 2IWIL does not show quite an advantage in some cases. The semi-supervised learning part may result in errors, thus introduces noise to the confidence prediction step. Also, if demonstrations are labeled with an inaccurate confidence score, the agent can only learn a non-optimal policy. Second, the objective function of 2IWIL is an estimation based on importance sampling, thus the error may inevitably occur.

To further investigate how the weight w works, we use $w = 1/D_\psi$ for comparison and we have 82.54 (Stage 1) and 155.85 (Stage 2) in Ant-v2 task. The result shows that simply taking $1/D_\psi$ as the weight achieves a decent performance. By contrast, our proposed weight not only investigates discriminator (i.e., D_ψ) but also the generator (i.e., π_θ), which enables a comprehensive examination of demonstrations.

Ratio of Optimal Demonstrations. Both "Stage 1" and "Stage 2" demonstrations are equally sampled from an optimal policy and a non-optimal policy. We consider "Stage 2" demonstrations in Ant-v2 and further study the performance of GAIL and WGAIL under different ratios of optimal demonstrations in Figure 2, where α denotes the ratio of optimal demonstrations. If the demonstrations are highly sub-optimal (e.g., $\alpha = \{0, 0.125\}$), both methods would be broken down. In robust ML, more than 50% contamination of data usually implies that we cannot distinguish the signal from the noise or outlier. In the simulation experiments, since the sub-optimal demonstrations are not absolute noise, we can still observe an increasingly large improvement over GAIL by exploiting the limited useful information when $\alpha = \{0.25, 0.5\}$. The advantage of WGAIL continues until $\alpha = 0.75$ and then WGAIL will become comparable with GAIL in the scenario of full-optimal demonstrations.

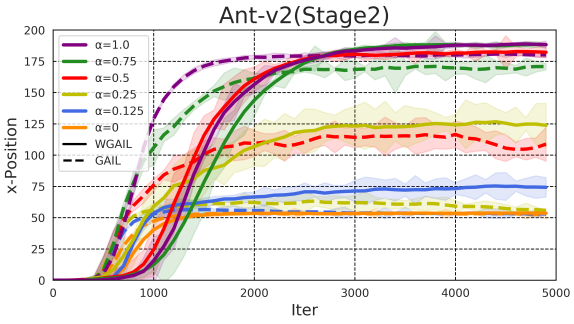

Figure 2. Performance of the learned agent with different ratios of optimal demonstrations.

Multiple Demonstrators. Consider "Stage 3" demonstrations are equally sampled from both 3 checkpoints and we conduct the experiment in Ant-v2 and HalfCheetah-v2 with "Stage 3" demonstrations to compare the performance of WGAIL and GAIL when dealing with demonstrations from multiple demonstrators. We have 172.62 (WGAIL) v.s. 122.96 (GAIL) on Ant-v2, 187.58 (WGAIL) v.s. 111.63 (GAIL) on HalfCheetah-v2, 14.43 (WGAIL) v.s. 7.16 (GAIL) on Walker2d-v2 and 15.56 (WGAIL) v.s. 14.29

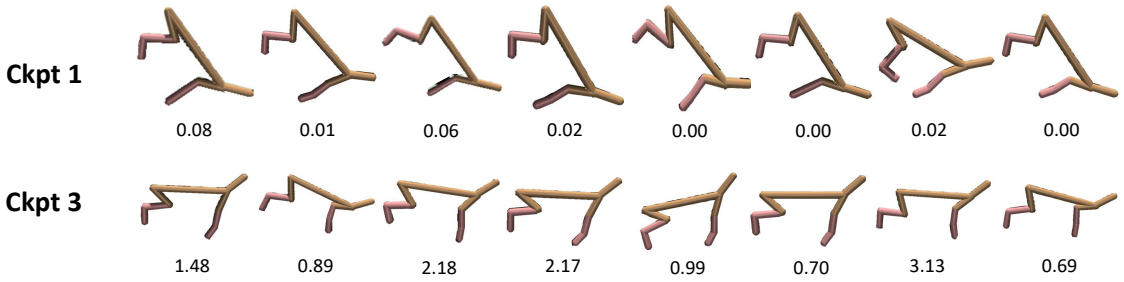

Figure 3. Visualized next state s' sampled from HalfCheetah-v2 task and weight w for (s, a).

(GAIL) on Hopper-v2. The result shows that WGAIL still works well when expert demonstrations are from multiple demonstrators.

5.1.2. Weight Visualization

In this subsection, we are going to examine the learned weight of each demonstration in the experiment. First, we report the final calculated average weight in the multiple demonstrators' experiment, as shown in Table 2.

	ckpt 1	ckpt 2	ckpt 3
Ant-v2	0.007	0.141	0.587
HalfCheetah-v2	0.092	0.296	1.163

Table 2. The average weight for demonstrations sampled from checkpoint 1-3.

As shown in Table 2, the weight can roughly refelct the sources of these demonstrations, e.g. demonstrations from checkpoint 3 usually have the largest weight. Then, we also visualize the Mujoco agent's body to show what the learned weight is like for each demonstration. The weight $w(s, a)$ is a function of state and action, however, the action of the agent can not be visualized intuitively. Consider a good state-action pair (s, a) can transform to a good next state s', we thus assign the next state s' to weight w to show the connection between the weight and the state-action pair, which means we suppose the agent will reach a good next state s' if the state-action pair (s, a) has a good weight w.

We calculate the weight in HalfCheetah-v2 task with "Stage 2" demonstrations. We sample two trajectories from checkpoint 1 and checkpoint 3 in HalfCheetah-v2 and then choose 8 triplets (s, a, s') to calculate the weight w and visualize s', as shown in Figure 3. Recall that checkpoint 3 is a better policy than checkpoint 1, thus we infer the weight of state-action pairs sampled from checkpoint 3 should be higher than that of checkpoint 1. Since the bad shape of the agent's body denotes the bad state of the agent, we can easily find that the lower weight is always connected to the worse state in checkpoint 1 and the better-visualized state in checkpoint 3 always has a higher weight in Figure 3.

	WGAIL	GAIL	BC	2IWIL	Expert
BeamRider	**1834.8**	1541.2	1034.5	1634.8	2637.45
Pong	-6.0	-6.7	-7.2	**9.9**	21.0
Q*bert	**15140.0**	-7.2	2720.0	4515.0	598.0
Seaquest	**649.0**	590.9	598.0	632.0	1840.0
Hero	20042.0	**20260.0**	7670.0	16990.0	27814.1

Table 3. The final result of the learned policy in five Atari tasks. The result is the final scores of average ten evaluations.

5.2. Experiment with Atari

Since the adversarial IRL method is generally hard to achieve good and stable performance on all Atari tasks (Tucker et al., 2018), we only evaluate WGAIL on five Atari games Beamrider, Pong, Qbert, Seaquest and Hero with one kind of imperfect demonstrations. To collect data, we use Proximal Policy Optimization (PPO) (Schulman et al., 2017) instead of TRPO to train an optimal Atari agent and the imperfect demonstrations are formed by two equivalent trajectories from two checkpoints during training. We use the Kostrikov's implementation of PPO (Kostrikov, 2018) and train with its default hyperparameter. Also, in the implementation of GAIL, we use PPO as the RL step instead of TRPO since it can outperform TRPO on Atari tasks and is faster in training.

The final result is summarized in Table 3 . The result shows that our new method WGAIL only has minor improvement than original GAIL in four out of five Atari games. Overall, both GAIL and WGAIL do not perform well. This illustrates that the weight estimation can partly help to the Atari agent training, however, there is an upper limit to reach expert-level performance for the GAN-based IRL methods in Atari games.

6. Conclusion

In this paper, we propose a new method for weighting imperfect expert demonstrations in imitation learning without auxiliary information. The weight of the demonstration acts as a measure to address the importance of each demonstration for agent training. Compared to existing related

algorithms, which generally require prior information of demonstrations, our method finds the connection between the weight estimation and plain GAIL problem, thus the weight can be calculated by the discriminator and agent policy during the training. We also give a detailed theoretical analysis to show an improved expert policy can be produced for agent learning. Experiment results in Mujoco and Atari domains show that the proposed method performs better than other baseline methods when handling imperfect demonstrations.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 61822113, the Australian Research Council under Projects DE180101438 and DP210101859.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1. ACM, 2004.

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

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Bain, M. and Sammut, C. A framework for behavioural cloning. In *Machine Intelligence 15*, pp. 103–129, 1995.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Brown, D. S., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. *arXiv preprint arXiv:1904.06387*, 2019.

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, 2020.

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

Codevilla, F., Miiller, M., López, A., Koltun, V., and Dosovitskiy, A. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–9. IEEE, 2018.

Englert, P., Paraschos, A., Peters, J., and Deisenroth, M. P. Model-based imitation learning by probabilistic trajectory matching. In *2013 IEEE International Conference on Robotics and Automation*, pp. 1922–1927. IEEE, 2013.

Ermon, S., Xue, Y., Toth, R., Dilkina, B., Bernstein, R., Damoulas, T., Clark, P., DeGloria, S., Mude, A., Barrett, C., et al. Learning large-scale dynamic discrete choice models of spatio-temporal preferences with application to migratory pastoralism in east africa. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Fang, T., Lu, N., Niu, G., and Sugiyama, M. Rethinking importance weighting for deep learning under distribution shift. *arXiv preprint arXiv:2006.04662*, 2020.

Finn, C., Levine, S., and Abbeel, P. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pp. 49–58, 2016.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Giusti, A., Guzzi, J., Cireşan, D. C., He, F.-L., Rodríguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., et al. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2015.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pp. 4565–4573, 2016.

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Ke, L., Barnes, M., Sun, W., Lee, G., Choudhury, S., and Srinivasa, S. Imitation learning as f-divergence minimization. *arXiv preprint arXiv:1905.12888*, 2019.

Kostrikov, I. Pytorch implementations of reinforcement learning algorithms. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018.

Kumar, M. P., Packer, B., and Koller, D. Self-paced learning for latent variable models. In *Advances in neural information processing systems*, pp. 1189–1197, 2010.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pp. 6389–6399, 2018.

Li, Y., Song, J., and Ermon, S. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*, pp. 3812–3822, 2017.

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. *arXiv preprint arXiv:1509.02971*, 2015.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pp. 278–287, 1999.

Oh, J., Guo, Y., Singh, S., and Lee, H. Self-imitation learning. *arXiv preprint arXiv:1806.05635*, 2018.

Peng, X. B., Kanazawa, A., Toyer, S., Abbeel, P., and Levine, S. Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. *arXiv preprint arXiv:1810.00821*, 2018.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 729–736. ACM, 2006.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343. PMLR, 2018.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015.

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

Stadie, B. C., Abbeel, P., and Sutskever, I. Third-person imitation learning. *arXiv preprint arXiv:1703.01703*, 2017.

Sugiyama, H., Meguro, T., and Minami, Y. Preference-learning based inverse reinforcement learning for dialog control. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tangkaratt, V., Han, B., Khan, M. E., and Sugiyama, M. Vild: Variational imitation learning with diverse-quality demonstrations. *arXiv preprint arXiv:1909.06769*, 2019.

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

Tucker, A., Gleave, A., and Russell, S. Inverse reinforcement learning for video games. *arXiv preprint arXiv:1810.10593*, 2018.

Valko, M., Ghavamzadeh, M., and Lazaric, A. Semi-supervised apprenticeship learning. In *European Workshop on Reinforcement Learning*, pp. 131–142, 2013.

Wirth, C., Furnkranz, J., Neumann, G., et al. Model-free preference-based reinforcement learning. In *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 2222–2228, 2016.

Wu, Y.-H., Charoenphakdee, N., Bao, H., Tangkaratt, V., and Sugiyama, M. Imitation learning from imperfect demonstration. *arXiv preprint arXiv:1901.09387*, 2019.

Xiao, H., Herman, M., Wagner, J., Ziesche, S., Etesami, J., and Linh, T. H. Wasserstein adversarial imitation learning. *arXiv preprint arXiv:1906.08113*, 2019.

Zheng, J., Liu, S., and Ni, L. M. Robust bayesian inverse reinforcement learning with sparse behavior noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.