A Ranking Game for Imitation Learning

Harshit Sikchi$^{\theta}$, **Akanksha Saran**$^{\phi}$, **Wonjoon Goo**$^{\theta}$, and **Scott Niekum**$^{\theta}$

$^{\theta}$Department of Computer Science, The University of Texas at Austin, USA
$^{\phi}$Microsoft Research NYC, USA
hsikchi@utexas.edu

Abstract

We propose a new framework for imitation learning—treating imitation as a *two-player ranking-based game* between a policy and a reward. In this game, the reward agent learns to satisfy pairwise performance rankings between behaviors, while the policy agent learns to maximize this reward. In imitation learning, near-optimal expert data can be difficult to obtain, and even in the limit of infinite data cannot imply a total ordering over trajectories as preferences can. On the other hand, learning from preferences alone is challenging as a large number of preferences are required to infer a high-dimensional reward function, though preference data is typically much easier to collect than expert demonstrations. The classical inverse reinforcement learning (IRL) formulation learns from expert demonstrations but provides no mechanism to incorporate learning from offline preferences and vice versa. We instantiate the proposed ranking-game framework with a novel ranking loss giving an algorithm that can simultaneously learn from expert demonstrations and preferences, gaining the advantages of both modalities. Our experiments show that the proposed method achieves state-of-the-art sample efficiency and can solve previously unsolvable tasks in the Learning from Observation (LfO) setting. Project code and details can be found at https://hari-sikchi.github.io/rank-game

1 Introduction

Reinforcement learning relies on environmental reward feedback to learn meaningful behaviors. Reward specification is a hard problem [1], thus motivating imitation learning (IL) as a technique to bypass reward specification and learn from expert data, often via Inverse Reinforcement Learning (IRL) techniques. Learning from expert information (imitation learning) alone can require efficient exploration when the expert actions are unavailable as in LfO [2]. Incorporating preferences over potentially suboptimal trajectories for reward learning can help reduce the exploration burden by regularizing the reward function and providing effective guidance for policy optimization. Previous literature in learning from preferences either assumes no environment interaction [3, 4] or assumes an active query framework with a restricted reward class [5]. The classical IRL formulation suffers from two issues: (1) Learning from expert demonstrations and learning from preferences/rankings provide complementary advantages for increasing learning efficiency [5, 6]; however, existing IRL methods that learn from expert demonstrations provide no mechanisms to incorporate offline preferences and vice versa. (2) Optimization is difficult, making learning sample inefficient [7, 8] due to the adversarial min-max game.

Our primary contribution is an algorithmic framework casting imitation learning as a ranking game that addresses both of the above issues in IRL. This framework treats imitation as a ranking game between two agents: a reward agent and a policy agent—the reward agent learns to satisfy pairwise performance rankings between different *behaviors* represented as state-action or state visitations, while the policy agent maximizes its performance under the learned reward function. The ranking game is detailed in Figure 1

IL Method	Offline Preferences	Expert Data	Ranking Loss	Reward Function	Active Human Query
MaxEntIRL, AdRIL,GAN-GCL, GAIL,f-MAX, AIRL	✗	LfD	supremum	non-linear	✗
BCO,GAIfO, DACfO, OPOLO,f-IRL	✗	LfO	supremum	non-linear	✗
TREX, DREX	✓	✗	Bradley-Terry	non-linear	✗
BREX	✓	✗	Bradley-Terry	linear	✗
DemPref	✓	LfO/LfD	Bradley-Terry	linear	✓
Ibarz et al[6]	✓	LfD	Bradley-Terry	non-linear	✓
`rank-game`	✓	LfO/LfD	L_k	non-linear	✗

Table 1: A summary of IL methods demonstrating the data modalities they can handle (expert data and/or preferences), the ranking-loss functions they use, the assumptions they make on reward function, and whether they require availability of an external agent to provide preferences during training. We highlight whether a method enables LfD, LfO, or both when it is able to incorporate expert data.

and is specified by three components: (1) The dataset of pairwise behavior rankings, (2) A ranking loss function, and (3) An optimization strategy. This game encompasses a large subset of both inverse reinforcement learning (IRL) methods and methods that learn from suboptimal offline preferences. Popular IRL methods such as GAIL, AIRL, f-MAX [8, 9, 10] are instantiations of this ranking game in which rankings are given only between the learning agent and the expert, and a gradient descent ascent (GDA) optimization strategy is used with a ranking loss that maximizes the performance gap between the behavior rankings.

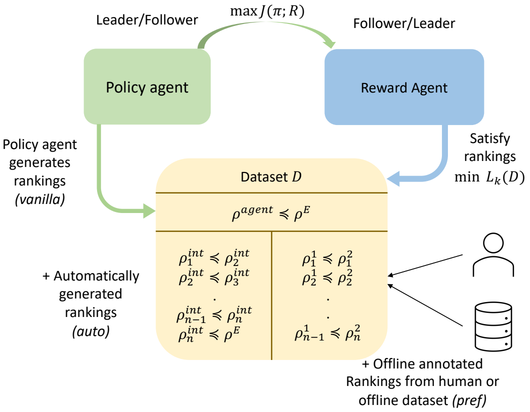
Leader/Follower
$\max J(\pi; R)$
Follower/Leader
Policy agent
Reward Agent
Policy agent generates rankings *(vanilla)*
Satisfy rankings $\min\ L_k(D)$
Dataset D
$\rho^{agent} \preccurlyeq \rho^E$
$\rho_1^{int} \preccurlyeq \rho_2^{int}$
$\rho_2^{int} \preccurlyeq \rho_3^{int}$
$\rho_{n-1}^{int} \preccurlyeq \rho_n^{int}$
$\rho_n^{int} \preccurlyeq \rho^E$
$\rho_1^1 \preccurlyeq \rho_1^2$
$\rho_2^1 \preccurlyeq \rho_2^2$
$\rho_{n-1}^1 \preccurlyeq \rho_n^2$
+ Automatically generated rankings *(auto)*
+ Offline annotated Rankings from human or offline dataset *(pref)*

Figure 1: `rank-game`: The Policy agent maximizes the reward function by interacting with the environment. The Reward agent satisfies a set of behavior rankings obtained from various sources: generated by the policy agent (vanilla), automatically generated (auto), or offline annotated rankings obtained from a human or offline dataset (pref). Treating this game in the Stackelberg framework leads to either Policy being a leader and Reward being a follower, or vice-versa.

The ranking loss used by the prior IRL approaches is specific to the comparison of optimal (expert) vs. suboptimal (agent) data and precludes the incorporation of comparisons among suboptimal behaviors. In this work, we instantiate the ranking game by proposing a new ranking loss (L_k) that facilitates incorporation of rankings over suboptimal trajectories for reward learning. Our theoretical analysis reveals that the proposed ranking loss results in a bounded performance gap with the expert that depends on a controllable hyperparameter. Our ranking loss can also ease policy optimization by supporting data augmentation to make the reward landscape smooth and allowing control over the learned reward scale. Finally, viewing our ranking game in the Stackelberg game framework (see Section 3)—an efficient setup for solving general-sum games—we obtain two algorithms with complementary benefits in non-stationary environments depending on which agent is set to be the leader.

In summary, this paper formulates a new framework `rank-game` for imitation learning that allows us to view learning from preferences and demonstrations under a unified perspective. We instantiate the framework with a principled ranking loss that can naturally incorporate rankings provided by diverse sources. Finally, by incorporating additional rankings—auto-generated or offline—our method: (a) outperforms state-of-the-art methods for imitation learning in several MuJoCo simulated domains by a significant margin and (b) solves complex tasks like imitating to reorient a pen with dextrous manipulation using only a few observation trajectories that none of the previous LfO baselines can solve.

2 Related Work

Imitation learning methods are broadly divided into two categories: Behavioral cloning [11, 12] and Inverse Reinforcement Learning (IRL) [8, 9, 13, 14, 15, 16, 17]. Our work focuses on developing a new framework in the setting of IRL through the lens of ranking. Table 1 shows a comparison of the proposed `rank-game` method to prior works.

Classical Imitation Game for IRL: The classical imitation game for IRL aims to solve the adversarial *min-max* problem of finding a policy that minimizes the worst-case performance gap between the agent and the expert. A number of previous works [9, 10, 18] have focused on analyzing the properties of this *min-max* game and its relation to divergence minimization. Under some additional regularization, this *min-max* objective can be understood as minimizing a certain f-divergence [8, 9, 10] between the agent and expert state-action visitation. More recently, [18] showed that all forms of imitation learning (BC and IRL) can be understood as performing moment matching under differing assumptions. In this work, we present a new perspective on imitation in which the reward function is learned using a dataset of behavior comparisons, generalizing previous IRL methods that learn from expert demonstrations and additionally giving the flexibility to incorporate rankings over suboptimal behaviors.

Learning from Preferences and Suboptimal Data: Learning from preferences and suboptimal data is important when expert data is limited or hard to obtain. Preferences [5, 19, 20, 21, 22] have the advantage of providing guidance in situations expert might not get into, and in the limit provides full ordering over trajectories which expert data cannot. A previous line of work [3, 4, 23, 24] has studied this setting and demonstrated that offline rankings over suboptimal behaviors can be effectively leveraged to learn a reward function. [5, 6, 22] studied the question of learning from preferences in the setting when a human is available to provide online preferences[1] (active queries), while [5] additionally assumed the reward to be linear in known features. Our work makes no such assumptions and allows for integrating offline preferences and expert demonstrations under a common framework.

Learning from Observation (LfO): LfO is the problem setting of learning from expert observations. This is typically more challenging than the traditional learning from demonstration setting (LfD), because actions taken by the expert are unavailable. LfO is broadly formulated using two objectives: state-next state marginal matching [25, 26, 27] and direct state marginal matching [28, 29]. Some prior works [30, 31, 32] approach LfO by inferring expert actions through a learned inverse dynamics model. These methods assume injective dynamics and suffer from compounding errors when the policy is deployed. A recently proposed method OPOLO [26] derives an upper bound for the LfO objective which enables it to utilize off-policy data and increase sample efficiency. Our method outperforms baselines including OPOLO, by a significant margin.

3 Background

We consider a learning agent in a Markov Decision Process (MDP) [33, 34] which can be defined as a tuple: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \rho_0)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces; P is the state transition probability function, with $P(s'|s, a)$ indicating the probability of transitioning from s to s' when taking action a; $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function bounded in $[0, R_{max}]$; We consider MDPs with infinite horizon, with the discount factor $\gamma \in [0, 1]$, though our results extend to finite horizons as well; p_0 is the initial state distribution. We use Π and $\mathcal{R}$ to denote the space of policies and reward functions respectively. A reinforcement learning agent aims to find a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes its expected return, $J(R; \pi) = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim\rho^\pi(s,a)}[R(s, a)]$, where $\rho^\pi(s, a)$ is the stationary state-action distribution induced by π. In imitation learning, we are provided with samples from the state-action visitation of the expert $\rho^{\pi_E}(s, a)$ but the reward function of the expert is unknown. We will use $\rho^E(s, a)$ as a shorthand for $\rho^{\pi_E}(s, a)$.

Classical Imitation Learning: The goal of imitation learning is to close the imitation gap $J(R; \pi^E) - J(R; \pi)$ defined with respect to the unknown expert reward function R. Several prior works [8, 18, 28, 35] tackle this problem by minimizing the imitation gap on all possible reward hypotheses. This leads to a zero-sum (min-max) game formulation of imitation learning in which a policy is optimized with respect to the reward function that induces the largest imitation gap:

$$\texttt{imit-game}(\pi) = \arg\min_{\pi\in\Pi} \sup_{f\in\mathcal{R}} \mathbb{E}_{\rho^E(s,a)}[f(s, a)] - \mathbb{E}_{\rho^\pi(s,a)}[f(s, a)]. \tag{1}$$

Here, the imitation gap is upper bounded as follows ($\forall\pi$):

$$J(R; \pi^E) - J(R; \pi) \leq \sup_{f\in\mathcal{R}} \mathbb{E}_{\rho^E(s,a)}[f(s, a)] - \mathbb{E}_{\rho^\pi(s,a)}[f(s, a)]. \tag{2}$$

[1] We will use preferences and ranking interchangebly

Note that, when the performance gap is maximized between the expert π^E and the agent π, we can observe that the worst-case reward function f_π induces a ranking between policy behaviors based on their performance: $\rho^E \succeq \rho^\pi := \mathbb{E}_{\rho^E(s,a)}[f_\pi(s,a)] \geq \mathbb{E}_{\rho^\pi(s,a)}[f_\pi(s,a)],\ \forall\pi$. Therefore, we can regard the above loss function that maximizes the performance gap (Eq. 2) as an instantiation of the ranking-loss. We will refer to the implicit ranking between the agent and the expert $\rho^E \succeq \rho^\pi$ as vanilla rankings and this variant of the ranking-loss function as the *supremum-loss*.

Stackelberg Games: A Stackelberg game is a general-sum game between two agents where one agent is set to be the leader and the other a follower. The leader in this game optimizes its objective under the assumption that the follower will choose the best response for its own optimization objective. More concretely, assume there are two players A and B with parameters θ_A, θ_B and corresponding losses $\mathcal{L}_A(\theta_A, \theta_B)$ and $\mathcal{L}_B(\theta_A, \theta_B)$. A Stackelberg game solves the following bi-level optimization when A is the leader and B is the follower: $\min_{\theta_A} \mathcal{L}_A(\theta_A, \theta_B^*(\theta_A))$ s.t $\theta_B^*(\theta_A) = \arg\min_\theta \mathcal{L}_B(\theta_A, \theta)$. [36] showed that casting model-based RL as an approximate Stackelberg game [37] leads to performance benefits and reduces training instability in comparison to the commonly used GDA [38] and Best Reponse (BR) [39] methods. [40, 41] prove convergence of Stackelberg games under smooth player cost functions and show that they reduce the cycling behavior to find an equilibrium and allow for better convergence.

4 A Ranking Game for Imitation Learning

In this section, we first formalize the notion of the proposed two-player general-sum ranking game for imitation learning. We then propose a practical instantiation of the ranking game through a novel ranking-loss (L_k). The proposed ranking game gives us the flexibility to incorporate additional rankings—both auto-generated (a form of data augmentation mentioned as 'auto' in Fig. 1) and offline ('pref' in Fig. 1)—which improves learning efficiency. Finally, we discuss the Stackelberg formulation for the two-player ranking game and discuss two algorithms that naturally arise depending on which player is designated as the leader.

4.1 The Two-Player Ranking Game Formulation

We present a new framework, `rank-game`, for imitation learning which casts it as a general-sum *ranking game* between two players — a reward and a policy.

$$\underbrace{\text{argmax}_{\pi\in\Pi} J(R;\pi)}_{\text{Policy Agent}} \qquad \underbrace{\text{argmin}_{R\in\mathcal{R}} L(\mathcal{D}^p;R)}_{\text{Reward Agent}}$$

In this formulation, the policy agent maximizes the reward by interacting with the environment, and the reward agent attempts to find a reward function that satisfies a set of pairwise behavior rankings in the given dataset $\mathcal{D}^p$; a reward function satisfies these rankings if $\mathbb{E}_{\rho^{\pi^i}}[R(s,a)] \leq \mathbb{E}_{\rho^{\pi^j}}[R(s,a)]$, $\forall \rho^{\pi^i} \preceq \rho^{\pi^j} \in \mathcal{D}^p$, where $\rho^{\pi^i}, \rho^{\pi^j}$ can be state-action or state vistitations.

The dataset of pairwise behavior rankings $\mathcal{D}^p$ can be comprised of the implicit 'vanilla' rankings between the learning agent and the expert's policy behaviors ($\rho^\pi \preceq \rho^E$), giving us the classical IRL methods when a specific ranking loss function – *supremum-loss* is used [8, 9, 10]. If rankings are provided between trajectories, they can be reduced to the equivalent ranking between the corresponding state-action/state visitations. In the case when $\mathcal{D}^p$ comprises purely of offline trajectory performance rankings then, under a specific ranking loss function (*Luce-shepard*), the ranking game reduces to prior reward inference methods like T-REX [3, 4, 23, 24]. Thus, the ranking game affords us a broader perspective of imitation learning, going beyond only using expert demonstrations.

4.2 Ranking Loss L_k for the Reward Agent

We use a *ranking-loss* to train the reward function—an objective that minimizes the distortion [42] between the ground truth ranking for a pair of entities $\{x, y\}$ and rankings induced by a parameterized function $R : \mathcal{X} \to \mathbb{R}$ for a pair of scalars $\{R(x), R(y)\}$. One type of such a ranking-loss is the *supremum-loss* in the classical imitation learning setup.

Algorithm 1 Meta algorithm: `rank-game` (vanilla) for imitation

```
Initialize policy $\pi_\theta^0$, reward funtion $R_\phi$, empty dataset $\mathcal{D}^\pi$. empirical expert data $\hat{\rho}^E$
for $t = 1..T$ iterations do
    Collect empirical visitation data $\hat{\rho}^{\pi_\theta^t}$ with $\pi_\theta^t$ in the environment. Set $\mathcal{D}^\pi = \{(\hat{\rho}^\pi \preceq \hat{\rho}^E)\}$
    Train reward $R_\phi$ to satisfy rankings in $\mathcal{D}^\pi$ using ranking loss $L_k$ in equation 3.
    Optimize policy under the reward function: $\pi_\theta^{t+1} \leftarrow \text{argmax}_{\pi'} J(R_\phi; \pi')$
end for
```

We propose a class of ranking-loss functions L_k that attempt to induce a performance gap of k for all behavior preferences in the dataset. Formally, this can be implemented with the regression loss:

$$L_k(\mathcal{D}^p; R) = \mathbb{E}_{(\rho^{\pi^i}, \rho^{\pi^j}) \sim \mathcal{D}^p} \Big[\mathbb{E}_{s,a\sim\rho^{\pi^i}} \big[(R(s,a) - 0)^2 \big] + \mathbb{E}_{s,a\sim\rho^{\pi^j}} \big[(R(s,a) - k)^2 \big] \Big]. \tag{3}$$

where $\mathcal{D}^p$ contains behavior pairs $(\rho^{\pi^i}, \rho^{\pi^j})$ s.t $\rho^{\pi^i} \preceq \rho^{\pi^j}$.

The proposed ranking loss allows for learning *bounded rewards with user-defined scale* k in the agent and the expert visitations as opposed to prior works in Adversarial Imitation Learning [8, 9, 17]. Reward scaling has been known to improve learning efficiency in deep RL; a large reward scale can make the optimization landscape less smooth [43, 44] and a small scale might make the action-gap small and increase susceptibility to extrapolation errors [45]. In contrast to the *supremum* loss, L_k can also naturally incorporate rankings provided by additional sources by learning a reward function satisfying all specified pairwise preferences. The following theorem characterizes the equilibrium of the `rank-game` for imitation learning when L_k is used as the ranking-loss.

Theorem 4.1. *(Performance of the `rank-game` equilibrium pair) Consider an equilibrium of the imitation `rank-game` $(\hat{\pi}, \hat{R})$, such that the ranking loss L_k generalization error is bounded by $2R_{max}^2\epsilon_r$ and the policy is near-optimal with $J(\hat{R}; \hat{\pi}) \geq J(\hat{R}; \pi) - \epsilon_\pi\ \forall \pi$, then at this equilibrium pair under the expert's unknown reward function R_{gt} bounded in $[0, R_{max}^E]$:*

$$\big| J(R_{gt}, \pi^E) - J(R_{gt}, \hat{\pi}) \big| \leq \frac{4R_{max}^E \sqrt{\frac{(1-\gamma)\epsilon_\pi + 4R_{max}\sqrt{\epsilon_r}}{k}}}{1-\gamma} \tag{4}$$

If reward is a state-only function and only expert observations are available, the same bound applies to the LfO setting.

Proof. We defer the proof to Appendix A. □

Theoretical properties: We now discuss some theoretical properties of L_k. Theorem 1 shows that `rank-game` has an equilibrium with bounded performance gap with the expert. An optimization step by the policy player, under a reward function optimized by the reward player, is equivalent to minimizing an f-divergence with the expert. Equivalently, at iteration t in Algorithm 1: $\max_{\pi^t} \mathbb{E}_{\rho^{\pi^t}}[R_t^*] - \mathbb{E}_{\rho^{\pi^E}}[R_t^*] = \min_{\pi^t} D_f(\rho^{\pi^t} \| \rho^{\pi^E})$. We elaborate on the regret of this idealized algorithm in Appendix A. Theorem 1 suggests that large values of k can guarantee the agent's performance is close to the expert. In practice, we observe intermediate values of k also preserve imitation equilibrium optimality with a benefit of promoting sample efficient learning (as an effect of reward scaling described earlier). We discuss this observation further in Appendix D.9. `rank-game` naturally extends to the LfO regime under a state-only reward function where Theorem 4.1 results in a divergence bound between state-visitations of the expert and the agent. A state-only reward function is also a sufficient and necessary condition to ensure that we learn a dynamics-disentangled reward function [17].

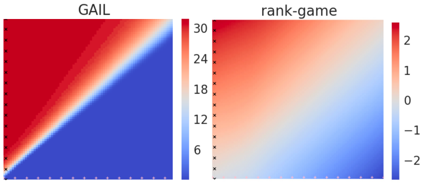

Figure 2: Figure shows learned reward function when agent and expert has a visitation shown by pink and black markers respectively. `rank-game` (auto) results in smooth reward functions more amenable to gradient-based policy optimization compared to GAIL.

L_k can incorporate additional preferences that can help learn a regularized/shaped reward function that provides better guidance for policy optimization, reducing the exploration burden and increasing sample efficiency for IRL. A better-guided policy optimization is also expected to incur a lower ϵ_π.

However, augmenting the ranking dataset can lead to a decrease in the intended performance gap ($k_{eff} < k$) between the agent and the expert (Appendix A). This can loosen the bound in Eq 4 and lead to non-optimal imitation learning. We hypothesize that given informative preferences, decreased ϵ_π can compensate for potentially decreased intended performance gap k_{eff} to ensure near optimal imitation. In our experiments, we observe this hypothesis holds true; we enjoy sample efficiency benefits without losing any asymptotic performance. To leverage these benefits, we present two methods for augmenting the ranking dataset below and defer the implementation details to Appendix B.

4.2.1 Augmenting the Ranking Dataset

Reward loss w/ automatically generated rankings (auto): In this method, we assume access to the behavior-generating trajectories in the ranking dataset. For each pairwise comparison $\rho_i \preceq \rho_j$ present in the dataset, L_k sets the regression targets for states in ρ_i to be 0 and for states visited by ρ_j to be k. Equivalently, we can rewrite minimizing L_k as regressing an input of trajectory τ_i to vector $\mathbf{0}$, and τ_j to vector $k\mathbf{1}$ where τ_i, τ_j are trajectories that generate the behavior ρ_i, ρ_j respectively. We use the comparison $\rho_i \preceq \rho_j$ to generate additional behavior rankings $\rho_i \preceq \rho_{\lambda_1,ij} \preceq \rho_{\lambda_2,ij} .. \preceq \rho_{\lambda_P,ij} \preceq \rho_j$ where $0 < \lambda_1 < \lambda_2 < ... < \lambda_P < 1$. The behavior $\rho_{\lambda_p,ij}$ is obtained by independently sampling the trajectories that generate the behaviors ρ_i, ρ_j and taking convex combinations i.e $\tau_{\lambda_p,ij} = \lambda_p\tau_i + (1-\lambda_p)\tau_j$ and their corresponding reward regressions targets are given by $\lambda_p\mathbf{0} + (1-\lambda_p)k\mathbf{1}$.

This form of data augmentation can be interpreted as mixup [46] regularization in the trajectory space. Mixup has been shown to improve generalization and adversarial robustness [46, 47] by regularizing the first and second-order gradients of the parameterized function. Following the general principle of using a smoothed objective with respect to inputs to obtain effective gradient signals, explicit smoothing in the trajectory space can also help reduce the policy optimization error ϵ_π. A didactic example showing rewards learned using this method is shown in Figure 2. In a special case when the expert's unknown reward function is linear in observations, these rankings reflect the true underlying rankings of behaviors.

Reward loss w/ offline annotated rankings (pref): Another way of increasing learning efficiency is augmenting the ranking dataset containing the vanilla ranking ($\rho^\pi \preceq \rho^E$) with offline annotated rankings. These rankings may be provided by a human observer or obtained using an offline dataset of behaviors with annotated reward information, similar to the datasets used in offline RL [48, 49]. We combine offline rankings by using a weighted loss between L_k for satisfying vanilla rankings ($\rho^\pi \preceq \rho^E$) and offline rankings, grounded by an expert. Providing offline rankings alone that are sufficient to explain the reward function of the expert [3] is often a difficult task and the number of offline preferences required depends on the complexity of the environment. In the LfO setting, learning from an expert's state visitation alone can be a hard problem due to exploration requirements [2]. This ranking-loss combines the benefits of using preferences to shape the reward function and guide policy improvement while using the expert to guarantee near-optimal performance.

4.3 Optimizing the Two-Player General-Sum Ranking Game as a Stackelberg Game

Solving the ranking-game in the Stackelberg setup allows us to propose two different algorithms depending on which agent is set to be the leader and utilize the learning stability and efficiency afforded by the formulation as studied in [36, 40, 41].

Policy as leader (PAL): Choosing policy as the leader implies the following optimization:

$$\max_\pi \left\{ J(\hat{R};\pi) \ \ s.t. \ \ \hat{R} = \arg\min_R L(\mathcal{D}^\pi; R) \right\} \tag{5}$$

Reward as leader (RAL): Choosing reward as the leader implies the following optimization:

$$\min_{\hat{R}} \left\{ L(\mathcal{D}^\pi; \hat{R}) \ \ s.t \ \ \pi = \arg\max_\pi J(\hat{R};\pi) \right\} \tag{6}$$

We follow the first order gradient approximation for leader's update from previous work [36] to develop practical algorithms. This strategy has been proven to be effective and avoids the computational complexity of calculating the implicit Jacobian term ($d\theta_B^*/d\theta_A$). PAL updates the reward to near convergence on dataset $\mathcal{D}^\pi$ ($\mathcal{D}^\pi$ contains rankings generated using the current policy agent only $\pi \preceq \pi^E$) and takes a few policy steps. Note that even after the first-order approximation, this optimization strategy differs from GDA as often only a few iterations are used for training the reward even in hyperparameter studies like [50]. RAL updates the reward conservatively. This is achieved through aggregating the dataset of implicit rankings from all previous policies obtained during training. PAL's strategy of using on-policy data $\mathcal{D}^\pi$ for reward training resembles that of methods including GAIL [8, 51], f-MAX [9], and f-IRL [28]. RAL uses the entire history of agent visitation to update the reward function and resembles methods such as apprenticeship learning and DAC [14, 52]. PAL and RAL bring together two seemingly different algorithm classes under a unified Stackelberg game viewpoint.

5 Experimental Results

We compare `rank-game` against state-of-the-art LfO and LfD approaches on MuJoCo benchmarks having continuous state and action spaces. The LfO setting is more challenging since no actions are available, and is a crucial imitation learning problem that can be used in cases where action modalities differ between the expert and the agent, such as in robot learning. We focus on the LfO setting in this section and defer the LfD experiments to Appendix D.2. We denote the imitation learning algorithms that use the proposed ranking-loss L_k from Section 4.2 as RANK-{PAL, RAL}. We refer to the `rank-game` variants which use automatically generated rankings and offline preferences as (auto) and (pref) respectively following Section 4.2. In all our methods, we rely on an off-policy model-free algorithm, Soft Actor-Critic (SAC) [53], for updating the policy agent.

We design experiments to answer the following questions:

1. *Asymptotic Performance and Sample Efficiency*: Is our method able to achieve near-expert performance given a limited number (1) of expert observations? Can our method learn using fewer environment interactions than prior state-of-the-art imitation learning (LfO) methods?
2. *Utility of preferences for imitation learning*: Current LfO methods struggle to solve a number of complex manipulation tasks with sparse success signals. Can we leverage offline annotated preferences through `rank-game` in such environments to achieve near-expert performance?
3. *Choosing between PAL and RAL methods*: Can we characterize the benefits and pitfalls of each method, and determine when one method is preferable over the other?
4. *Ablations for the method components*: Can we establish the importance of hyperparameters and design decisions in our experiments?

Baselines: We compare RANK-PAL and RANK-RAL against 6 representative LfO approaches that covers a spectrum of on-policy and off-policy model-free methods from prior work: GAIfO [8, 51], DACfO [52], BCO [30], f-IRL [28] and recently proposed OPOLO [26] and IQLearn [54]. We do not assume access to expert actions in this setting. Our LfD experiments compare to the IQLearn [54], DAC [52] and BC baselines. Detailed description for baselines can be found in Appendix D.2.

5.1 Asymptotic Performance and Sample Efficiency

In this section, we compare RANK-PAL(auto) and RANK-RAL(auto) to baselines on a set of MuJoCo locomotion tasks of varying complexities: `Swimmer-v2, Hopper-v2, HalfCheetah-v2, Walker2d-v2, Ant-v2` and `Humanoid-v2`. In this experiment, we provide one expert trajectory for all methods and do not assume access to any offline annotated rankings.

Env	Hopper	HalfCheetah	Walker	Ant	Humanoid
BCO	20.10±2.15	5.12±3.82	4.00±1.25	12.80±1.26	3.90±1.24
GaIFO	81.13± 9.99	13.54±7.24	83.83±2.55	20.10±24.41	3.93±1.81
DACfO	94.73±3.63	85.03±5.09	54.70±44.64	86.45±1.67	19.31±32.19
f -IRL	97.45± 0.61	96.06±4.63	**101.16±1.25**	71.18±19.80	77.93±6.372
OPOLO	89.56±5.46	88.92±3.20	79.19±24.35	93.37± 3.78	24.87±17.04
RANK-PAL(ours)	87.14± 16.14	94.05±3.59	93.88±0.72	**98.93±1.83**	**96.84±3.28**
RANK-RAL(ours)	**99.34±0.20**	**101.14±7.45**	93.24±1.25	93.21±2.98	94.45±4.13
Expert	100.00± 0	100.00± 0	100.00± 0	100.00± 0	100.00± 0
$(\lvert\mathcal{S}\rvert, \lvert\mathcal{A}\rvert)$	$(11, 3)$	$(17, 6)$	$(17, 6)$	$(111, 8)$	$(376, 17)$

Table 2: Asymptotic normalized performance of LfO methods at 2 million timesteps on MuJoCo locomotion tasks. The standard deviation is calculated with 5 different runs each averaging over 10 trajectory returns. For unnormalized score and more details, check Appendix D. We omit IQlearn due to poor performance.

Asymptotic Performance: Table 2 shows that both `rank-game` methods are able to reach near-expert asymptotic performance with a single expert trajectory. BCO shows poor performance which can be attributed to the compounding error problem arising from its behavior cloning strategy. GAIfO and DACfO use GDA for optimization with a supremum loss and show high variance in their asymptotic performance whereas `rank-game` methods are more stable and low-variance.

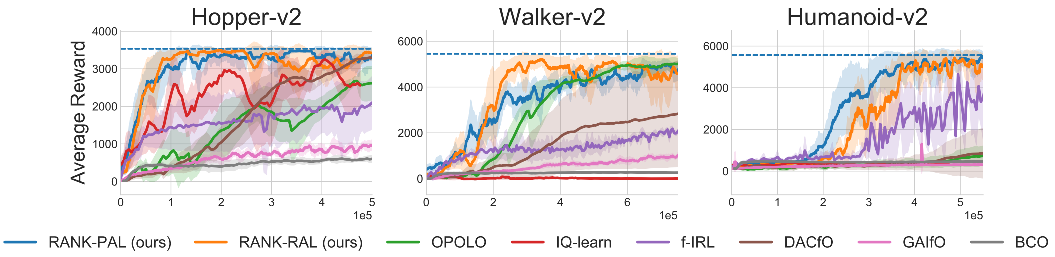

Figure 3: Comparison of performance on OpenAI gym benchmark tasks. The shaded region represents the standard deviation across 5 random runs. RANK-PAL and RANK-RAL substantially outperform the baselines in sample efficiency. Complete set of results can be found in Appendix D.1

Sample Efficiency: Figure 3 shows that RANK-RAL and RANK-PAL are among the most sample efficient methods for the LfO setting, outperforming the recent state-of-the-art method OPOLO [26] by a significant margin. We notice that IQLearn fails to learn in the LfO setting. This experiment demonstrates the benefit of the combined improvements of the proposed ranking-loss with automatically generated rankings. Our method is also simpler to implement than OPOLO, as we require fewer lines of code changes on top of SAC and need to maintain fewer parameterized networks compared to OPOLO which requires an additional inverse action model to regularize learning.

5.2 Utility of Preferences in Imitation

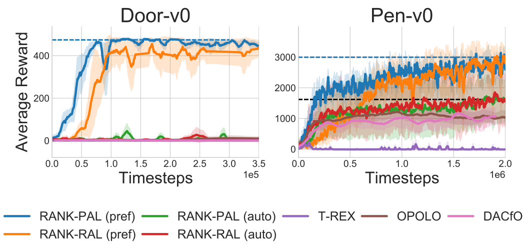

Figure 4: Offline annotated preferences can help solve LfO tasks in the complex manipulation environments Pen-v0 and Door, whereas prior LfO methods fail. Black dotted line shows the asymptotic performance of RANK-PAL (auto) method.

Our experiments on complex manipulation environments—door opening with a parallel-jaw gripper [55] and pen manipulation with a dexterous adroit hand [56] – reveal that none of the prior LfO methods are able to imitate the expert even under increasing amounts of expert data. This failure of LfO methods can be potentially attributed to the exploration requirements of LfO compared to LfD [2], coupled with the sparse successes encountered in these tasks, leading to poorly guided policy gradients. In these experiments, we show that `rank-game` can incorporate additional information in the form of offline annotated rankings to guide the agent in solving such tasks. These offline rankings are obtained by uniformly sampling a small set of trajectories (10) from the replay buffer of SAC [53] labeled with a ground truth reward function. We use a weighted ranking loss (pref) from Section 4.2.

Figure 4 shows that RANK-PAL/RAL(pref) method leveraging offline ranking is the only method that can solve these tasks, whereas prior LfO methods and RANK-PAL/RAL(auto) with automatically generated rankings struggle even after a large amount of training. We also point out that T-REX, a method that learns using the preferences alone is unable to achieve near-expert performance, thereby highlighting the benefits of learning from expert demonstrations alongside a set of offline preferences.

5.3 Comparing PAL and RAL

PAL uses the agent's current visitation for reward learning, whereas RAL learns a reward consistent with all rankings arising from the history of the agent's visitation. These properties can present certain benefits depending on the task setting. To test the potential benefits of PAL and RAL, we consider two non-stationary imitation learning problems, similar to [56] – one in which the expert changes it's intent and the other where dynamics of the environment change during training in the Hopper-v2 locomotion task. For changing intent, we present a new set of demonstrations where the hopper agent hops backward rather than forward. For changing environment dynamics, we increase the mass of the hopper agent by a factor of 1.2. Changes are introduced at 1e5 time steps during training at which point we notice a sudden performance drop.

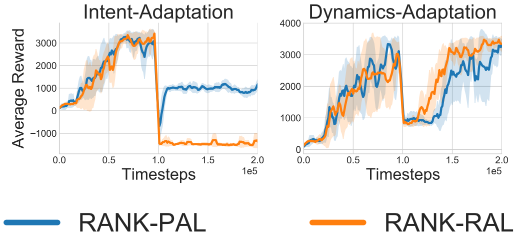

Figure 5: We compare the relative strengths of PAL and RAL. Left plot shows a comparison when the goal is changed, and right plot shows a comparison when the dynamics of the environment is changed. These changes occur at 1e5 timesteps into training. PAL adapts faster to changing intent and RAL adapts faster to changing dynamics.

In Figure 5 (left), we notice that PAL adapts faster to intent changes, whereas RAL needs to unlearn the rankings obtained from the agent's history and takes longer to adapt. Figure 5 (right) shows that RAL adapts faster to the changing dynamics of the system, as it has already learned a good global notion of the dynamics-disentangled reward function in the LfO setting, whereas PAL only has a local understanding of reward as a result of using ranking obtained only from the agent's current visitation.

Ablation of Method Components: Appendix D contains eight additional experiments to study the importance of hyperparameters and design decisions. Our ablations validate the importance of using automatically generated rankings, the benefit of ranking loss over *supremum* loss, and sensitivity to hyperparameters like the intended performance gap k, policy iterations, and the reward regularizer.

6 Conclusion

In this work, we present a new framework for imitation learning that treats imitation as a two-player ranking-game between a policy and a reward function. Unlike prior works in imitation learning, the ranking game allows incorporation of rankings over suboptimal behaviors to aid policy learning. We instantiate the ranking game by proposing a novel ranking loss which guarantees agent's performance to be close to expert for imitation learning. Our experiments on simulated MuJoCo tasks reveal that utilizing additional ranking through our proposed ranking loss leads to improved sample efficiency for imitation learning, outperforming prior methods by a significant margin and solving some tasks which were unsolvable by previous LfO methods.

Limitations and Negative Societal Impacts: Preferences obtained in the real world are usually noisy [57, 58, 59] and one limitation of `rank-game` is that it does not suggest a way to handle noisy preferences. Second, `rank-game` proposes modifications to learn a reward function amenable to policy optimization but these hyperparameters are set manually. Future work can explore methods to automate learning such reward functions. Third, despite learning effective policies we observed that we do not learn reusable robust reward functions [28]. Negative Societal Impact: Imitation learning can cause harm if given demonstrations of harmful behaviors, either accidentally or purposefully. Furthermore, even when given high-quality demonstrations of desirable behaviors, our algorithm does not provide guarantees of performance and thus could cause harm if used in high-stakes domains without sufficient safety checks on learned behaviors.

7 Acknowledgements

We thank Jordan Schneider, Wenxuan Zhou, Caleb Chuck, and Amy Zhang for valuable feedback on the paper. This work has taken place in the Personal Autonomous Robotics Lab (PeARL) at The University of Texas at Austin. PeARL research is supported in part by the NSF (IIS-1724157, IIS-1749204, IIS-1925082), AFOSR (FA9550-20-1-0077), and ARO (78372-CS). This research was also sponsored by the Army Research Office under Cooperative Agreement Number W911NF-19-2-0333. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] V. Krakovna, “Specification gaming examples in ai,” *Available at vkrakovna. wordpress. com*, 2018.

[2] R. Kidambi, J. Chang, and W. Sun, “Mobile: Model-based imitation learning from observation alone,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[3] D. S. Brown, W. Goo, P. Nagarajan, and S. Niekum, “Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations,” *ArXiv*, vol. abs/1904.06387, 2019.

[4] D. Brown, R. Coleman, R. Srinivasan, and S. Niekum, “Safe imitation learning via fast bayesian reward inference from preferences,” in *International Conference on Machine Learning*, pp. 1165–1177, PMLR, 2020.

[5] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, “Learning reward functions by integrating human demonstrations and preferences,” *arXiv preprint arXiv:1906.08928*, 2019.

[6] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei, “Reward learning from human preferences and demonstrations in atari,” *arXiv preprint arXiv:1811.06521*, 2018.

[7] O. Arenz and G. Neumann, “Non-adversarial imitation learning and its connections to adversarial methods,” *arXiv preprint arXiv:2008.03525*, 2020.

[8] J. Ho and S. Ermon, “Generative adversarial imitation learning,” *Advances in neural information processing systems*, vol. 29, pp. 4565–4573, 2016.

[9] S. K. S. Ghasemipour, R. Zemel, and S. Gu, “A divergence minimization perspective on imitation learning methods,” in *Conference on Robot Learning*, pp. 1259–1277, PMLR, 2020.

[10] L. Ke, S. Choudhury, M. Barnes, W. Sun, G. Lee, and S. Srinivasa, “Imitation learning as f-divergence minimization,” in *Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14*, pp. 313–329, Springer International Publishing, 2021.

[11] D. A. Pomerleau, “Efficient training of artificial neural networks for autonomous navigation,” *Neural computation*, vol. 3, no. 1, pp. 88–97, 1991.

[12] S. Ross, G. Gordon, and D. Bagnell, “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, JMLR Workshop and Conference Proceedings, 2011.

[13] A. Y. Ng, S. J. Russell, *et al.*, “Algorithms for inverse reinforcement learning.,” in *Icml*, vol. 1, p. 2, 2000.

[14] P. Abbeel and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning,” in *Proceedings of the twenty-first international conference on Machine learning*, p. 1, 2004.

[15] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, “Maximum entropy inverse reinforcement learning.,” in *Aaai*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

[16] C. Finn, S. Levine, and P. Abbeel, “Guided cost learning: Deep inverse optimal control via policy optimization,” in *International conference on machine learning*, pp. 49–58, PMLR, 2016.

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

[18] G. Swamy, S. Choudhury, J. A. Bagnell, and S. Wu, “Of moments and matching: A game-theoretic framework for closing the imitation gap,” in *International Conference on Machine Learning*, pp. 10022–10032, PMLR, 2021.

[19] R. Akrour, M. Schoenauer, and M. Sebag, “Preference-based policy learning,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 12–27, Springer, 2011.

[20] A. Wilson, A. Fern, and P. Tadepalli, “A bayesian approach for policy learning from trajectory preference queries,” *Advances in neural information processing systems*, vol. 25, pp. 1133–1141, 2012.

[21] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, “Active preference-based learning of reward functions,” 2017.

[22] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” *arXiv preprint arXiv:1706.03741*, 2017.

[23] D. S. Brown, W. Goo, and S. Niekum, “Better-than-demonstrator imitation learning via automatically-ranked demonstrations,” in *Conference on robot learning*, pp. 330–359, PMLR, 2020.

[24] L. Chen, R. Paleja, and M. Gombolay, “Learning from suboptimal demonstration via self-supervised reward regression,” *arXiv preprint arXiv:2010.11723*, 2020.

[25] F. Torabi, G. Warnell, and P. Stone, “Recent advances in imitation learning from observation,” *arXiv preprint arXiv:1905.13566*, 2019.

[26] Z. Zhu, K. Lin, B. Dai, and J. Zhou, “Off-policy imitation learning from observations,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[27] W. Sun, A. Vemula, B. Boots, and D. Bagnell, “Provably efficient imitation learning from observation alone,” in *International conference on machine learning*, pp. 6036–6045, PMLR, 2019.

[28] T. Ni, H. Sikchi, Y. Wang, T. Gupta, L. Lee, and B. Eysenbach, “f-irl: Inverse reinforcement learning via state marginal matching,” *arXiv preprint arXiv:2011.04709*, 2020.

[29] F. Liu, Z. Ling, T. Mu, and H. Su, “State alignment-based imitation learning,” *arXiv preprint arXiv:1911.10947*, 2019.

[30] F. Torabi, G. Warnell, and P. Stone, “Behavioral cloning from observation,” *arXiv preprint arXiv:1805.01954*, 2018.

[31] C. Yang, X. Ma, W. Huang, F. Sun, H. Liu, J. Huang, and C. Gan, “Imitation learning from observations by minimizing inverse dynamics disagreement,” *arXiv preprint arXiv:1910.04417*, 2019.

[32] A. Edwards, H. Sahni, Y. Schroecker, and C. Isbell, “Imitating latent policies from observation,” in *International Conference on Machine Learning*, pp. 1755–1763, PMLR, 2019.

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

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

[35] I. Kostrikov, O. Nachum, and J. Tompson, “Imitation learning via off-policy distribution matching,” *arXiv preprint arXiv:1912.05032*, 2019.

[36] A. Rajeswaran, I. Mordatch, and V. Kumar, “A game theoretic framework for model based reinforcement learning,” in *ICML*, 2020.

[37] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[38] F. Schäfer and A. Anandkumar, “Competitive gradient descent,” *arXiv preprint arXiv:1905.12103*, 2019.

[39] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[40] T. Fiez, B. Chasnov, and L. J. Ratliff, “Convergence of learning dynamics in stackelberg games,” *arXiv preprint arXiv:1906.01217*, 2019.

[41] L. Zheng, T. Fiez, Z. Alumbaugh, B. Chasnov, and L. J. Ratliff, "Stackelberg actor-critic: Game-theoretic reinforcement learning algorithms," *arXiv preprint arXiv:2109.12286*, 2021.

[42] R. Iyer and J. Bilmes, "The submodular bregman and lovász-bregman divergences with applications: Extended version," Citeseer, 2012.

[43] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[44] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, JMLR Workshop and Conference Proceedings, 2010.

[45] M. G. Bellemare, G. Ostrovski, A. Guez, P. Thomas, and R. Munos, "Increasing the action gap: New operators for reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[46] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[47] H. Guo, Y. Mao, and R. Zhang, "Mixup as locally linear out-of-manifold regularization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3714–3722, 2019.

[48] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.

[49] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[50] M. Orsini, A. Raichuk, L. Hussenot, D. Vincent, R. Dadashi, S. Girgin, M. Geist, O. Bachem, O. Pietquin, and M. Andrychowicz, "What matters for adversarial imitation learning?," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[51] F. Torabi, G. Warnell, and P. Stone, "Generative adversarial imitation from observation," *arXiv preprint arXiv:1807.06158*, 2018.

[52] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson, "Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning," *arXiv preprint arXiv:1809.02925*, 2018.

[53] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.

[54] D. Garg, S. Chakraborty, C. Cundy, J. Song, and S. Ermon, "Iq-learn: Inverse soft-q learning for imitation," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[55] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.

[56] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[57] M. Kwon, E. Biyik, A. Talati, K. Bhasin, D. P. Losey, and D. Sadigh, "When humans aren't optimal: Robots that collaborate with risk-aware humans," in *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 43–52, IEEE, 2020.

[58] H. J. Jeon, S. Milli, and A. Dragan, "Reward-rational (implicit) choice: A unifying formalism for reward learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4415–4426, 2020.

[59] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences," *The International Journal of Robotics Research*, p. 02783649211041652, 2021.

[60] A. Rényi, "On measures of entropy and information," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pp. 547–561, University of California Press, 1961.

[61] S. M. Ali and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 28, no. 1, pp. 131–142, 1966.

[62] I. Csiszár, "Information-type measures of difference of probability distributions and indirect observation," *studia scientiarum Mathematicarum Hungarica*, vol. 2, pp. 229–318, 1967.

[63] F. Liese and I. Vajda, "On divergences and informations in statistics and information theory," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4394–4412, 2006.

[64] T. Xu, Z. Li, and Y. Yu, "Error bounds of imitating policies and environments for reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[65] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[66] I. Vajda, "Note on discrimination information and variation (corresp.)," *IEEE Transactions on Information Theory*, vol. 16, no. 6, pp. 771–773, 1970.

[67] G. Gilardoni, "On the minimum f-divergence for given total variation," *Comptes Rendus Mathematique - C R MATH*, vol. 343, pp. 763–766, 12 2006.

[68] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The collected works of Wassily Hoeffding*, pp. 409–426, Springer, 1994.

[69] M. Kearns and S. Singh, "Finite-sample convergence rates for q-learning and indirect algorithms," *Advances in neural information processing systems*, vol. 11, 1998.

[70] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *In Proc. 19th International Conference on Machine Learning*, Citeseer, 2002.

[71] S. Reddy, A. D. Dragan, and S. Levine, "Sqil: Imitation learning via reinforcement learning with sparse rewards," *arXiv preprint arXiv:1905.11108*, 2019.

[72] J. Achiam, "Spinning Up in Deep Reinforcement Learning," 2018.