Pareto Inverse Reinforcement Learning for Diverse Expert Policy Generation

Woo Kyung Kim, **Minjong Yoo**, **Honguk Woo***
Department of Computer Science and Engineering, Sungkyunkwan University
{kwk2696, mjyoo2, hwoo}@skku.edu

Abstract

Data-driven offline reinforcement learning and imitation learning approaches have been gaining popularity in addressing sequential decision-making problems. Yet, these approaches rarely consider learning Pareto-optimal policies from a limited pool of expert datasets. This becomes particularly marked due to practical limitations in obtaining comprehensive datasets for all preferences, where multiple conflicting objectives exist and each expert might hold a unique optimization preference for these objectives. In this paper, we adapt inverse reinforcement learning (IRL) by using reward distance estimates for regularizing the discriminator. This enables progressive generation of a set of policies that accommodate diverse preferences on the multiple objectives, while using only two distinct datasets, each associated with a different expert preference. In doing so, we present a Pareto IRL framework (ParIRL) that establishes a Pareto policy set from these limited datasets. In the framework, the Pareto policy set is then distilled into a single, preference-conditioned diffusion model, thus allowing users to immediately specify which expert's patterns they prefer. Through experiments, we show that ParIRL outperforms other IRL algorithms for various multi-objective control tasks, achieving the dense approximation of the Pareto frontier. We also demonstrate the applicability of ParIRL with autonomous driving in CARLA.

1 Introduction

In decision-making scenarios, each expert might have her own preference on multiple, possibly conflicting objectives (multi-objectives). Accordingly, learning Pareto-optimal policies in multi-objective environments has been considered essential and practical to provide users with a selection of diverse expert-level policies, which can cater their specific preferences (e.g., [Xu *et al.*, 2020; Kyriakis *et al.*, 2022]). However, in the area of imitation learning, such multi-objective problem has not been fully explored due to the requirement for comprehensive expert datasets encompassing the full range of multi-objective preferences (e.g., [Zhu *et al.*, 2023]), which might be unattainable in real-world scenarios.

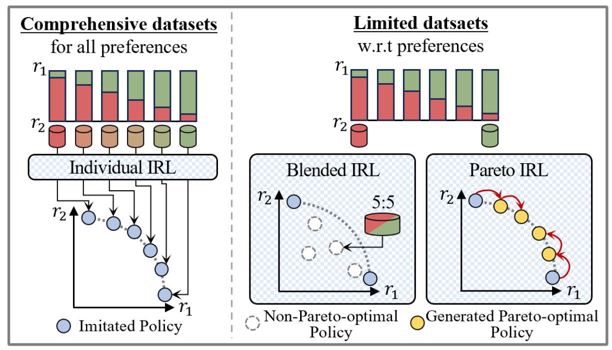


Figure 1: Data-driven Pareto policy set learning

In the ideal scenario depicted on the left side of Figure 1, having comprehensive expert datasets encompassing diverse multi-objective preferences enables the straightforward derivation of a Pareto policy set by reconstructing policies from each dataset. However, this is often not feasible in real-world situations where datasets might not represent all preferences. This common limitation is illustrated on the right side of Figure 1. Here, one typically has access to only two distinct datasets, each reflecting different multi-objective preferences. In such limited dataset cases, a viable approach involves merging these datasets in varying proportions, followed by the application of imitation learning on each blended dataset. However, this approach often leads to a collection of non-Pareto-optimal policies, as demonstrated in Section 4.

In this paper, we address the challenges of multi-objective imitation learning in situations with strictly limited datasets, specifically focusing on Pareto policy set generation. Our goal is to derive optimal policies that conform with diverse multi-objective preferences, even in the face of limited datasets regarding these preferences. To do so, we investigate inverse reinforcement learning (IRL) and present a Pareto IRL (ParIRL) framework in which a Pareto policy set corresponding to the best compromise solutions over multi-objectives can be induced. This framework is set in a simi-

*Honguk Woo is the corresponding author.

lar context to conventional IRL where reward signals are not from the environment, but it is intended to obtain a dense set of Pareto policies rather than an individually imitated policy.

In ParIRL, we exploit a recursive IRL structure to find a Pareto policy set progressively in a way that at each step, nearby policies can be derived between the policies of the previous step. Specifically, we adapt IRL using reward distance regularization; new policies are regularized based on reward distance estimates to be balanced well between distinct datasets, while ensuring the regret bounds of each policy. This recursive IRL is instrumental in achieving the dense approximation of a Pareto policy set. Through distillation of the approximated Pareto policy set to a single policy network, we build a diffusion-based model, which is conditioned on multi-objective preferences. This distillation not only enhances the Pareto policy set but also integrates it into a single unified model, thereby facilitating the zero-shot adaptation to varying and unseen preferences.

The contributions of our work are summarized as follows.

- We introduce the ParIRL framework to address a novel challenge of imitation learning, Pareto policy set generation from strictly limited datasets.
- We devise a recursive IRL scheme with reward distance regularization to generate policies that extend beyond the datasets, and we provide a theoretical analysis on their regret bounds.
- We present a preference-conditioned diffusion model to further enhance the approximated policy set on unseen preferences. This allows users to dynamically adjust their multi-objective preferences at runtime.
- We verify ParIRL with several multi-objective environments and autonomous driving scenarios, demonstrating its superiority for Pareto policy set generation.
- ParIRL is the first to tackle the data limitation problem for Pareto policy set generation within the IRL context.

2 Preliminaries and Problem Formulation

2.1 Background

Multi-Objective RL (MORL). A multi-objective Markov decision process (MOMDP) is formulated with multiple reward functions, each associated with an individual objective.

$$(S,\ A,\ P,\ \mathbf{r},\ \Omega,\ f, \gamma) \tag{1}$$

Here, $s \in S$ is a state space, $a \in A$ is an action space, $P : S \times A \times S \to [0, 1]$ is a transition probability, and $\gamma \in [0, 1]$ is a discount factor. MOMDP incorporates a vector of m reward functions $\mathbf{r} = [r_1, ..., r_m]$ for $r : S \times A \times S \to \mathbb{R}$, a set of preference vectors $\Omega \subset \mathbb{R}^m$, and a linear preference function $f(\mathbf{r}, \boldsymbol{\omega}) = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{r}$ where $\boldsymbol{\omega} \in \Omega$. The goal of MORL is to find a set of Pareto polices $\pi^* \in \Pi^*$ for an MOMDP environment, where π^* maximizes scalarized returns, i.e., $\max_\pi \mathbb{E}_{a \sim \pi(\cdot|s)}[\sum_{t=1}^{H} \gamma^t f(\boldsymbol{r}, \boldsymbol{\omega})]$.

Inverse RL (IRL). Given an expert dataset $\mathcal{T}^* = \{\tau_i\}_{i=1}^n$, where each trajectory τ_i is represented as a sequence of state and action pairs $\{(s_t, a_t)\}_{t=1}^T$, IRL aims to infer the reward function of the expert policy, thus enabling the rationalization of its behaviors. Among many, the adversarial IRL algorithm (AIRL) casts IRL into a generative adversarial problem [Fu *et al.*, 2018; Wang *et al.*, 2024] with such discriminator as

$$D(s, a, s') = \frac{\exp(\tilde{r}(s, a, s'))}{\exp(\tilde{r}(s, a, s')) + \pi(a|s)} \tag{2}$$

where $s' \sim P(s, a, \cdot)$ and $\tilde{r}$ is a inferring reward function. The discriminator is trained to maximize the cross entropy between expert dataset and dataset induced by the policy via

$$\begin{aligned} \max\ [\mathbb{E}_{(s,a)\sim\mathcal{T}_\pi}[\log(1 - D(s, a, s'))]+ \\ \mathbb{E}_{(s,a)\sim\mathcal{T}^*}[\log D(s, a, s')]] \end{aligned} \tag{3}$$

where $\mathcal{T}_\pi$ is the dataset induced by learning policy π. The generator of AIRL corresponds to π, which is trained to maximize the entropy-regularized reward function such as

$$\begin{aligned} \log(D(s, a, s')) - \log(1 - D(s, a, s')) \\ = \tilde{r}(s, a, s') - \log \pi(a|s). \end{aligned} \tag{4}$$

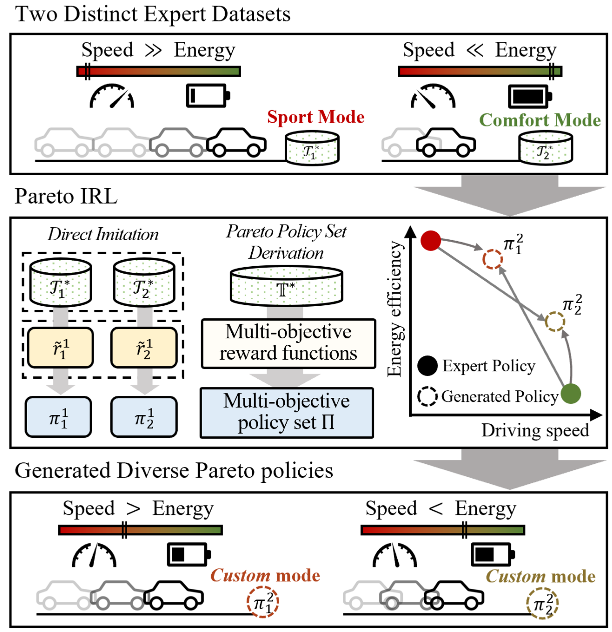


Figure 2: Concept of Pareto policy set generation: given two distinct expert datasets, each associated with a specific preference over multi-objectives (e.g., some expert prefers speed over energy efficiency, and vice versa), the Pareto IRL is to find a set of optimal compromise *custom* policies, each of which can conform to a different preference.

2.2 Formulation of Pareto IRL

We specify the Pareto IRL problem which derives a Pareto policy set from strictly limited datasets. Consider M distinct expert datasets $\mathbb{T}^* = \{\mathcal{T}_i^*\}_{i=1}^M$ where each expert dataset $\mathcal{T}_i^*$ is collected from the optimal policy on some reward function $r_{\text{mo}} = \boldsymbol{\omega}_i^{\mathrm{T}}\mathbf{r}$ with a fixed preference $\boldsymbol{\omega}_i \in \Omega$. Furthermore, we assume that each dataset $\mathcal{T}_i^*$ distinctly exhibits dominance on a particular reward function r_i. In the following, we consider scenarios with two objectives ($M = 2$), and later discuss the generalization for three or more objectives in Appendix A.4.

Given two distinct datasets, in the context of IRL, we refer to Pareto policy set derivation via IRL as Pareto IRL. Specifically, it aims at inferring a reward function $\tilde{r}$ and learning a policy π for any preference $\boldsymbol{\omega}$ from the strictly limited datasets $\mathbb{T}^*$. That is, when exploiting limited expert datasets in a multi-objective environment, we focus on establishing the Pareto policy set effectively upon unknown reward functions and preferences.

Figure 2 briefly illustrates the concept of Pareto IRL, where a self-driving task involves different preferences on two objectives, possibly conflicting, such as driving speed and energy efficiency. Consider two distinct expert datasets, where each expert has her own preference settings for the driving speed and energy efficiency objectives (e.g., $\mathcal{T}_1^*$ and $\mathcal{T}_2^*$ involve one dominant objective differently). While it is doable to restore a single useful policy individually from one given expert dataset, our work addresses the issue to generate a set of policies Π which can cover a wider range of preferences beyond given datasets. The policies are capable of rendering optimal compromise returns, denoted by dotted circles in the figure, and they allow users to immediately select the optimal solution according to their preference and situation.

For an MOMDP with a set of preference vectors $\boldsymbol{\omega} \in \Omega$, a vector of reward functions $\mathbf{r}$, and a preference function f in (1), Pareto policy set generation is to find a set of multi-objective policies such as

$$\Pi = \{\pi \mid R_{f(\mathbf{r},\boldsymbol{\omega})}(\pi) \geq R_{f(\mathbf{r},\boldsymbol{\omega})}(\pi'), \forall \pi', \exists\, \boldsymbol{\omega} \in \Omega\} \quad (5)$$

for M expert preference datasets $\{\mathcal{T}_i^*\}_{i=1}^M$. $R_r(\pi)$ represents returns induced by policy π on reward function r. Neither a vector of true reward functions $\mathbf{r}$ is explicitly revealed, nor the rewards signals are annotated in the expert datasets, similar to conventional IRL scenarios.

3 Our Framework

To obtain a Pareto policy set from strictly limited datasets, we propose the ParIRL framework involving two learning phases: (i) recursive reward distance regularized IRL, (ii) distillation to a preference-conditioned model.

In the first phase, our approach begins with direct imitation of the given expert datasets, and then recursively finds neighboring policies that lie on the Pareto front. Specifically, we employ the reward distance regularized IRL method that incorporates reward distance regularization into the discriminator's objective to learn a robust multi-objective reward function. This regularized IRL ensures that the performance of the policy learned by the inferred multi-objective reward function remains within the bounds of the policy learned by the true reward function. By performing this iteratively, we achieve new useful policies that are not presented in the expert datasets, thus establishing a high-quality Pareto policy set.

In the second phase, we distill the Pareto policy set into a preference-conditioned diffusion model. The diffusion model encapsulates both preference-conditioned and unconditioned policies, each of which is associated with the preference-specific knowledge (within a task) and the task-specific knowledge (across all preferences), respectively. Consequently, the unified policy model further enhances the Pareto policy set, rendering robust performance for arbitrary *unseen* preferences in a zero-shot manner. It also allows for efficient resource utilization with a single policy network.

3.1 Recursive Reward Distance Regularized IRL

Notation. We use superscripts $g \in \{1, ..., G\}$ to denote recursive step and subscripts $i \in \{1, 2\}$ to denote i-th multi-objective policies derived at each recursive step g. We consider two objectives cases in the following.

Individual IRL. As shown in the Figure 3 (i-1), the framework initiates with two separate IRL procedures, each dedicated to directly imitating one of the expert datasets. For this, we adopt AIRL [Fu *et al.*, 2018] which uses the objectives (3) and (4) to infer reward functions $\{\tilde{r}_i^1\}_{i=1}^2$ and policies $\{\pi_i^1\}_{i=1}^2$ from the individual expert dataset $\mathcal{T}_i^* \in \mathbb{T}^*$.

Reward distance regularized IRL. Subsequently, as shown in the Figure 3 (i-2), at each recursive step $g \geq 2$, we derive new multi-objective reward functions $\{\tilde{r}_i^g\}_{i=1}^2$ and respective policies $\{\pi_i^g\}_{i=1}^2$ that render beyond the given datasets. To do so, a straightforward approach might involve conducting IRL iteratively by blending the expert datasets at different ratios. However, as illustrated in Figure 4(a), the resulting policies tend to converge towards some weighted mean of datasets, rather than fully exploring non-dominant optimal actions beyond simple interpolation of given expert actions.

To address the problem, we present a reward distance regularized IRL on datasets $\mathbb{T}^{g-1} = \{\mathcal{T}_i^{g-1}\}_{i=1}^2$ collected from the policies derived at the previous step. Given a reward distance metric $d(r, r')$, we compute the distance between the newly derived reward function $\tilde{r}_i^g$ and previously derived reward functions $\tilde{\mathbf{r}}^{g-1} = [\tilde{r}_1^{g-1}, \tilde{r}_2^{g-1}]$. Further, we define target distances as a vector $\boldsymbol{\epsilon}_i^g = [\epsilon_{i,1}^g, \epsilon_{i,2}^g]$ to constrain each of the corresponding measured reward distances. Then, we define a reward distance regularization term as

$$I(\tilde{r}_i^g, \tilde{\mathbf{r}}^{g-1}) = \sum_{j=1}^2 \left(\epsilon_{i,j}^g - d(\tilde{r}_i^g, \tilde{r}_j^{g-1})\right)^2 \quad (6)$$

where the subscripts i and j denote the newly derived reward function and the previously derived one, respectively. Finally, we incorporate (6) into the discriminator objective (3) as

$$\begin{aligned}\max\ & \mathbb{E}_{(s,a)\sim\mathcal{T}_{\pi_i^g}}[\log(1 - D(s, a, s'))] + \\ & \mathbb{E}_{(s,a)\sim\mathbb{T}^{g-1}}[\log D(s, a, s')] - \beta \cdot I(\tilde{r}_i^g, \tilde{\mathbf{r}}^{g-1})\end{aligned} \quad (7)$$

where β is a hyperparameter. This allows the discriminator to optimize a multi-objective reward function for a specific target distance across datasets. The reward distance regularized IRL procedure is performed twice with different target distances, to derive policies adjacent to each of the previously derived policies. Furthermore, we fork the new regularized IRL procedure with the previously one (that is adjacent) to enhance the efficiency and robustness in learning.

The choice of the target distance is crucial, as the regret of a multi-objective policy is bounded under the reward distance (11). Thus, we set the sum of the target distances as small as possible. As the reward distance metrics satisfy the triangle inequality $d(\tilde{r}_1^{g-1}, \tilde{r}_2^{g-1}) \leq d(\tilde{r}_i^g, \tilde{r}_1^{g-1}) +$

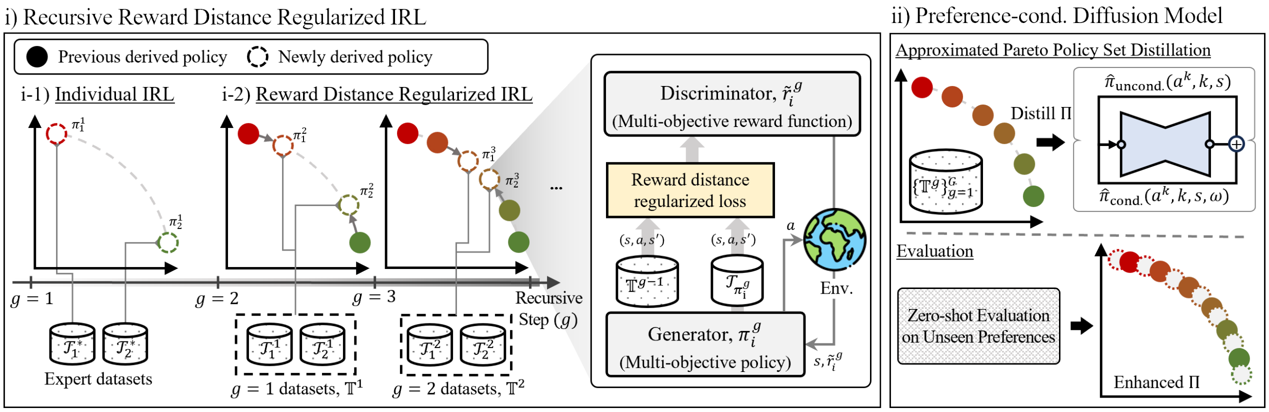


Figure 3: ParIRL framework: In (i), policies in a Pareto set are recursively derived via reward distance regularized IRL. In (ii), the preference-conditioned diffusion model enhances the approximated Pareto policy set via distillation.

Algorithm 1 Recursive reward distance regularized IRL

Input: Expert preference datasets $\mathbb{T}^* = \{\mathcal{T}_1^*, \mathcal{T}_2^*\}, \Pi = \emptyset$

```
/* 1-st step: individual IRL procedures */
for i ← 1, ..., 2 do
    Obtain r̃_i^1 and π_i^1 using T_i^* through (3) and (4)
    Collect dataset T_i^1 by executing π_i^1
    Execute 𝕋^1 ← 𝕋^1 ∪ T_i^1 and Π ← Π ∪ {π_i^1}
/* g-th step: regularized IRL procedure */
for g ← 2, ..., G do
    for i ← 1, ..., 2 do
        Initialize r̃_i^g ← r̃_i^{g−1} and π_i^g ← π_i^{g−1}
        Set ε_i^g = [ε_{i,1}^g, ε_{i,2}^g] based on (8)
        while not converge do
            Update r̃_i^g using ε_i^g and regularized loss in (7)
            Update π_i^g w.r.t. r̃_i^g using (4)
        Collect dataset T_i^g by executing π_i^g
        Execute 𝕋^g ← 𝕋^g ∪ T_i^g and Π ← Π ∪ π_i^g
return Π
```

$d(\tilde{r}_i^g, \tilde{r}_2^{g-1})$, we limit the sum of target distances to

$$\hat{\epsilon}_i^g = \sum_{j=1}^{2} \epsilon_{i,j}^g = d(\tilde{r}_1^{g-1}, \tilde{r}_2^{g-1}). \tag{8}$$

In practice, we assign a small constant value for one of the target distances $\epsilon_{i,i}^g$, while the other is determined as $\hat{\epsilon}_i^g - \epsilon_{i,i}^g$. By doing so, we are able to effectively derive a new policy that is adjacent to one of the previous policies.

Any reward distance metric that guarantees the regret bounds of policy can be used for ParIRL. In our implementation, we adopt EPIC, also known as equivalent policy invariant comparison pseudometric [Gleave *et al.*, 2020], which quantitatively measures the distance between two reward functions. The learning procedure of recursive reward distance regularized IRL is summarized in Algorithm 1. In Appendix A.4, we discuss the generalization of reward distance regularization for more than two objectives ($M \geq 3$).

3.2 Regret Bounds of Reward Distance Regularized Policy

We provide an analysis of the regret bounds of a reward distance regularized policy. Let $\tilde{r}$ be our learned reward function and π_r^* be the optimal policy with respect to reward function r. Suppose that there exists a (ground truth) multi-objective reward function $r_{\text{mo}} = \boldsymbol{\omega}^T \mathbf{r}$ with preference $\boldsymbol{\omega} = [\omega_1, \omega_2]$. With the linearity of r_{mo}, we obtain

$$\begin{aligned} R_{r_{\text{mo}}}(\pi_{r_{\text{mo}}}^*) - R_{r_{\text{mo}}}(\pi_{\tilde{r}}^*) &= \sum_{i=1}^{2} \omega_i (R_{\tilde{r}_i}(\pi_{r_{\text{mo}}}^*) - R_{\tilde{r}_i}(\pi_{\tilde{r}}^*)) \\ &\leq \sum_{i=1}^{2} \omega_i (R_{\tilde{r}_i}(\pi_{\tilde{r}_i}^*) - R_{\tilde{r}_i}(\pi_{\tilde{r}}^*)). \end{aligned} \tag{9}$$

Let $\mathcal{D}$ be the distribution over transitions $S \times A \times S$ used to compute EPIC distance d_ϵ, and $\mathcal{D}_{\pi,t}$ be the distribution over transitions on timestep t induced by policy π. Using Theorem A.16 in [Gleave *et al.*, 2020], we derive that for $\alpha \geq 2$, (9) is bounded by the sum of individual regret bounds, i.e.,

$$\begin{aligned} &\sum_{i=1}^{2} \omega_i (R_{\tilde{r}_i}(\pi_{\tilde{r}_i}^*) - R_{\tilde{r}_i}(\pi_{\tilde{r}}^*)) \\ &\leq \sum_{i=1}^{2} 16 \omega_i \|\tilde{r}_i\|_2 \left(K d_\epsilon(\tilde{r}, \tilde{r}_i) + L \Delta_\alpha(\tilde{r})\right) \end{aligned} \tag{10}$$

where L is a constant, $K = \alpha/(1-\gamma)$, $\Delta_\alpha(\tilde{r}) = \sum_{t=0}^{T} \gamma^t W_\alpha(\mathcal{D}_{\pi_{\tilde{r}}^*,t}, \mathcal{D})$, and W_α is the relaxed Wasserstein distance [Villani, 2003]. Consequently, we obtain

$$\begin{aligned} &R_{r_{\text{mo}}}(\pi_{r_{\text{mo}}}^*) - R_{r_{\text{mo}}}(\pi_{\tilde{r}}^*) \\ &\leq 32K \|r_{\text{mo}}\|_2 \left(\sum_{i=1}^{2} [\omega_i d_\epsilon(\tilde{r}, \tilde{r}_i)] + \frac{L}{K} \Delta_\alpha(\tilde{r}) \right). \end{aligned} \tag{11}$$

As such, the regret bounds of our learned policy π on reward function $\tilde{r}$ are represented by the regularization term based on EPIC along with the differences between the respective distributions of transitions generated by $\pi_{\tilde{r}}^*$ and the distribution $\mathcal{D}$ used to compute EPIC distance. This ensures that the regret bounds of π can be directly optimized by using (7). In our implementation, instead of directly multiplying the preference $\boldsymbol{\omega}$ to the loss function, we reformulate the preference into the target distance to balance the distance better. The details with proof can be found in Appendix A.2.

Environment	Metric	Oracle	DiffBC	BeT	GAIL	AIRL	IQ-Learn	DiffAIL	ParIRL	ParIRL+DU
MO-Car	HV	6.06	3.95 ± 0.26	4.22 ± 0.06	4.95 ± 0.16	5.01 ± 0.13	3.47 ± 1.20	1.78 ± 0.08	5.37 ± 0.08	$\mathbf{5.89 \pm 0.05}$
	SP	0.01	1.49 ± 0.19	1.07 ± 0.23	0.89 ± 0.43	0.67 ± 0.10	3.43 ± 2.84	6.95 ± 0.12	0.26 ± 0.05	$\mathbf{0.01 \pm 0.01}$
	CR	1.00	0.81 ± 0.01	0.82 ± 0.04	0.60 ± 0.09	0.83 ± 0.02	0.69 ± 0.03	0.69 ± 0.07	$\mathbf{0.97 \pm 0.01}$	$\mathbf{0.97 \pm 0.01}$
MO-Swimmer	HV	5.60	3.56 ± 0.38	3.86 ± 0.20	3.61 ± 0.13	3.83 ± 0.38	2.97 ± 0.35	3.52 ± 1.03	4.56 ± 0.04	$\mathbf{4.96 \pm 0.06}$
	SP	0.03	1.12 ± 0.24	0.74 ± 0.34	1.80 ± 0.34	2.34 ± 1.34	2.54 ± 1.31	4.00 ± 4.62	0.17 ± 0.02	$\mathbf{0.01 \pm 0.01}$
	CR	1.00	0.75 ± 0.01	0.82 ± 0.05	0.70 ± 0.09	0.70 ± 0.01	0.78 ± 0.03	0.77 ± 0.07	$\mathbf{0.98 \pm 0.01}$	0.96 ± 0.01
MO-Cheetah	HV	5.09	3.97 ± 0.27	2.22 ± 0.29	3.75 ± 0.28	4.25 ± 0.06	2.82 ± 0.48	3.86 ± 0.46	4.97 ± 0.13	$\mathbf{5.27 \pm 0.07}$
	SP	0.01	0.59 ± 0.23	1.26 ± 0.95	1.56 ± 0.50	0.62 ± 0.13	6.68 ± 3.51	1.40 ± 0.83	0.11 ± 0.01	$\mathbf{0.01 \pm 0.00}$
	CR	1.00	0.72 ± 0.02	0.68 ± 0.04	0.72 ± 0.03	0.77 ± 0.03	0.55 ± 0.04	0.38 ± 0.10	0.92 ± 0.02	$\mathbf{0.93 \pm 0.02}$
MO-Ant	HV	6.30	3.73 ± 0.23	1.90 ± 0.09	3.63 ± 0.22	4.08 ± 0.29	2.34 ± 0.28	1.40 ± 0.01	4.71 ± 0.10	$\mathbf{4.99 \pm 0.12}$
	SP	0.01	0.33 ± 0.13	0.61 ± 0.79	0.67 ± 0.13	0.37 ± 0.16	0.58 ± 0.24	15.09 ± 0.01	0.06 ± 0.01	$\mathbf{0.01 \pm 0.00}$
	CR	1.00	0.87 ± 0.00	0.73 ± 0.03	0.86 ± 0.02	0.91 ± 0.01	0.69 ± 0.02	0.19 ± 0.03	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.99 \pm 0.01}$
MO-AntXY	HV	6.78	3.76 ± 0.04	1.54 ± 0.12	4.18 ± 0.16	4.49 ± 0.19	2.02 ± 0.06	3.66 ± 0.63	5.37 ± 0.09	$\mathbf{5.61 \pm 0.10}$
	SP	0.03	0.54 ± 0.11	12.14 ± 0.88	0.50 ± 0.02	0.39 ± 0.14	0.70 ± 0.13	1.06 ± 0.24	0.07 ± 0.01	$\mathbf{0.01 \pm 0.00}$
	CR	1.00	0.80 ± 0.02	0.42 ± 0.06	0.76 ± 0.03	0.77 ± 0.04	0.68 ± 0.02	0.52 ± 0.16	0.95 ± 0.02	$\mathbf{0.98 \pm 0.00}$
MO-Car* (3-obj)	HV	2.89	0.86 ± 0.12	1.54 ± 0.00	1.63 ± 0.02	1.88 ± 0.09	0.80 ± 0.02	1.16 ± 0.07	1.87 ± 0.04	$\mathbf{2.79 \pm 0.02}$
	SP	0.01	0.46 ± 0.12	0.06 ± 0.00	0.08 ± 0.01	0.90 ± 0.74	0.53 ± 0.03	0.19 ± 0.02	0.08 ± 0.00	$\mathbf{0.01 \pm 0.00}$

Table 1: Performance of Pareto set generation: regarding evaluation metrics, the higher HV, the higher the performance; the lower SP, the higher the performance; the higher CR, the higher the performance. For the baselines and ParIRL, we evaluate with 3 random seeds.

3.3 Preference-conditioned Diffusion Model

To further enhance the Pareto policy set Π obtained in the previous section, we leverage diffusion models [HO *et al.*, 2020; Ho and Salimans, 2022], interpolating and extrapolating policies via distillation. We first systematically annotate Π with preferences $\boldsymbol{\omega} \in \Omega$ in an ascending order. We then train a diffusion-based policy model, which is conditioned on these preferences; i.e.,

$$\pi_{\text{u}}(a|s,\omega) = \mathcal{N}(a^K; 0, I) \prod_{k=1}^{K} \hat{\pi}_{\text{u}}(a^{k-1}|a^k, k, s, \omega) \quad (12)$$

where superscripts $k \sim [1, K]$ denote the denoising timestep, $a^0 (= a)$ is the original action, and a^{k-1} is a marginally denoised version of a^k. The diffusion model is designed to predict the noise from a noisy input $a^k = \sqrt{\bar{\alpha}^k} a + \sqrt{1 - \bar{\alpha}^k} \eta$ with a variance schedule parameter $\bar{\alpha}^k$ and $\eta \sim \mathcal{N}(0, I)$, i.e.

$$\min \mathop{\mathbb{E}}_{(s,a)\sim\{\mathbb{T}^g\}_{g=1}^G, k\sim[1,K]} [||\hat{\pi}_{\text{u}}(a^k, k, s, \omega) - \eta||_2^2] \quad (13)$$

where $\{\mathbb{T}^g\}_{g=1}^G$ is the entire datasets collected by the policies in Π. Furthermore, we represent the model as a combination of preference-conditioned and unconditioned policies,

$$\begin{aligned} &\hat{\pi}_{\text{u}}(a^k, k, s, \omega) := \\ &\quad (1 - \delta) \hat{\pi}_{\text{cond.}}(a^k, k, s, \omega) + \delta \hat{\pi}_{\text{uncond.}}(a^k, k, s) \end{aligned} \quad (14)$$

where δ is a guidance weight. The unconditioned policy encompasses general knowledge across the approximated Pareto policies, while the conditioned one guides the action according to the specific preference.

During sampling, the policy starts from a random noise and iteratively denoises it to obtain the executable action,

$$a^{k-1} = \frac{1}{\sqrt{\alpha^k}} \left(a^k - \frac{1 - \alpha^k}{\sqrt{1 - \bar{\alpha}^k}} \hat{\pi}_{\text{u}}(a^k, k, s, \omega) \right) + \sigma^k \eta \quad (15)$$

where α^k and σ^k are variance schedule parameters. The diffusion model $\hat{\pi}_{\text{u}}$ allows for efficient resource utilization at runtime with a single policy network, and is capable of rendering robust performance for *unseen* preferences in a zero-shot manner. Consequently, it enhances the Pareto policy set in terms of Pareto front density , as illustrated in Figure 3 (ii).

4 Evaluation

4.1 Experiment Settings

Environments. For evaluation, we use (i) a multi-objective car environment (MO-Car), and several multi-objective variants of MuJoCo environments used in the MORL literature [Xu *et al.*, 2020; Kyriakis *et al.*, 2022] including (ii) MO-Swimmer, (iii) MO-Cheetah, (iv) MO-Ant, and (v) MO-AntXY. For tradeoff objectives, the forward speed and the energy efficiency are used in (ii)-(iv), and the x-axis speed and the y-axis speed are used in (v). In these environments, similar to conventional IRL settings, reward signals are not used for training; they are used solely for evaluation.

Baselines. For comparison, we implement following imitation learning algorithms: 1) **DiffBC** [Pearce *et al.*, 2023], an imitation learning method that uses a diffusion model for the policy, 2) **BeT** [Shafiullah *et al.*, 2022], an imitation learning method that integrates action discretization into the transformer architecture, 3) **GAIL** [Ho and Ermon, 2016], an imitation learning method that imitates expert dataset via the generative adversarial framework, 4) **AIRL** [Fu *et al.*, 2018], an IRL method that induces both the reward function and policy, 5) **IQ-Learn** [Garg *et al.*, 2021], an IRL method that learns a q-function to represent both the reward function and policy, 6) **DiffAIL** [Wang *et al.*, 2024], an IRL method that incorporates the diffusion loss to the discriminator's objective. To cover a wide range of different preferences, these baselines are conducted multiple times on differently augmented datasets, where each is a mixed dataset that integrates given datasets in the same ratio to a specific preference. We also include MORL [Xu *et al.*, 2020] that uses explicit rewards from the environment, unlike IRL settings. It serves as **Oracle** (the upper bound of performance) in the comparison.

Metrics. For evaluation, we use several multi-objective metrics [Yang *et al.*, 2019; Xu *et al.*, 2020].

- Hypervolume metric (HV) represents the quality in the cumulative returns of a Pareto policy set. Let $\mathcal{F}$ be the Pareto frontier obtained from an approximated Pareto policy set for m objectives and $\boldsymbol{R}_0 \in \mathbb{R}^m$ be a reference point for each objective. Then, HV $= \int \mathbb{1}_{H(\mathcal{F})}(z)dz$ where $H(\mathcal{F}) = \{z \in \mathbb{R}^m \mid \exists \boldsymbol{R} \in \mathcal{F} : \boldsymbol{R_0} \leq z \leq \boldsymbol{R}\}$.
- Sparsity metric (SP) represents the density in the average return distance of the Pareto frontier. Let $\mathcal{F}_j(i)$ be the i-th value in a sorted list for the j-th objective. Then, SP $= \frac{1}{|\mathcal{F}|-1}\sum_{j=1}^{m}\sum_{i=1}^{|\mathcal{F}|}(\mathcal{F}_j(i) - \mathcal{F}_j(i+1))^2$.

We also use a new metric designed for Pareto IRL.

- Coherence metric (CR) represents the monotonic improvement property of approximated policy set $\Pi = \{\pi_i\}_{i\leq N}$ generated by two expert datasets. Let policy list $(\pi_1, ..., \pi_N)$ be sorted in ascending order by the expected return of the policies with respect to reward function r_1, Then, CR $= \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i}^{N}\mathbb{1}_{h(i,j)}$ where $h(i,j)$=$R_{r_1}(\pi_i)$≤$R_{r_1}(\pi_j)$ and $R_{r_2}(\pi_i)$≥$R_{r_2}(\pi_j)$.

For HV and CR, higher is better, but for SP, lower is better.

4.2 Performance of Pareto Set Generation

Table 1 compares the performance in the evaluation metrics (HV, SP, CR) achieved by our framework (ParIRL, ParIRL+DU) and other baselines (DiffBC, BeT, GAIL, AIRL, IQ-Learn, DiffAIL). ParIRL is trained with the recursive reward distance regularized IRL, and ParIRL+DU is enhanced through the distillation. For the baselines, the size of a preference set (with different weights) is given equally to the number of policies derived via ParIRL. When calculating HV and SP, we exclude the out-of-order policies obtained from an algorithm with respect to preferences. As shown, our ParIRL and ParIRL+DU consistently yield the best performance for all environments, outperforming the most competitive baseline AIRL by $15.6\% \sim 23.7\%$ higher HV, $80.4\% \sim 98.2\%$ lower SP, and $21.7\% \sim 22.2\%$ higher CR on average. Furthermore, we observe an average HV gap of 9.8% between ParIRL+DU and Oracle that uses the ground truth reward signals. This gap is expected, as existing IRL algorithms are also known to experience a performance drop compared to RL algorithms that directly use reward signals [Fu *et al.*, 2018]. For the baselines, such performance degradation is more significant, showing an average drop of 26.9% in HV between AIRL and Oracle. ParIRL+DU improves the performance in HV over ParIRL by 7.0% on average, showing the distilled diffusion model achieves robustness on unseen preferences.

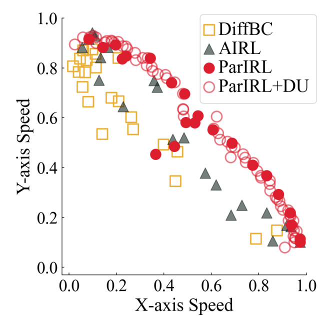
To verify the performance of ParIRL for three objectives case, we extend MO-Car to MO-Car* where the tradeoff objectives are the velocities on three different directions. Our ParIRL and ParIRL+DU show superiority in terms of HV, but sometimes show slightly lower performance in SP. I t is because the baselines tend to shrinks towards the low-performance region, thus yielding lower SP. As CR is defined only for two objectives cases, CR for MO-Car* is not reported. The generalization of reward distance regularization for three or more objectives is discussed in Appendix A.4.

In this experiment, the baselines exhibit relatively low performance due to their primarily concentration on imitating the datasets, posing a challenge in generating policies that go beyond the limited datasets. Specifically, as DiffBC and BeT are designed to handle datasets with multiple modalities, they do not necessarily lead to the generation of novel actions. Meanwhile, the IRL baselines demonstrate relatively better performance, as they involve environment interactions. However, imitating from a merged dataset with specific ratio tends to converge towards the mean of existing actions, thus leading to sub-optimal performance.

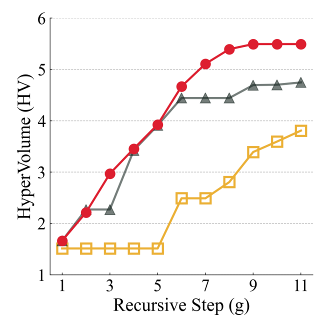
4.3 Analysis

Pareto Visualization. Figure 4(a) depicts the Pareto policy set by our ParIRL and ParIRL+DU as well as the baselines (DiffBC, AIRL) for MO-AntXY. The baselines often produce the non-optimal solutions, specified by the dots in the low-performance region. ParIRL+DU produces the most densely spread policies, which lie on the high-performance region.

Learning Efficiency. Figure 4(b) depicts the learning curves in HV for MO-AntXY over recursive steps. For baselines, we intentionally set the number of policies of the baselines equal to the number of policies derived through ParIRL for each step. The curves show the superiority of our recursive reward distance regularized IRL in generating the higher quality (HV) Pareto frontier. Furthermore, the recursive learning scheme significantly reduces the training time, requiring only $13\% \sim 25\%$ of training timesteps compared to the IRL baselines. This is because ParIRL explores adjacent policies progressively by making explicit use of the previously derived policies to fork another regularized IRL procedure.



(a) Pareto Visualization



(b) Learning Curve

Figure 4: Pareto policy set Π visualization and learning curve

Ablation Studies. Table 2 provides an ablation study of ParIRL with respect to the reward distance metrics and recursive learning scheme. For this, we implement ParIRL/MSE and ParIRL/PSD, which use mean squared error (MSE) and Pearson distance (PSD) for reward distance measures, respectively; we also implement ParIRL/RC which represents ParIRL without recursive learning scheme. While MSE tends to compute the exact reward distance and PSD estimates the linear correlation between rewards, EPIC accounts for the reward function distance that is invariant to potential shaping [Gleave *et al.*, 2020], thus making ParIRL optimize the regret bounds of a policy learned on an inferred reward function. Moreover, ParIRL/RC degrades compared to ParIRL, clarifying the benefit of our recursive learning scheme.

Env.	Met.	ParIRL/MSE	ParIRL/PSD	ParIRL/RC	ParIRL
1	HV	3.37 ± 0.08	4.02 ± 0.23	4.17 ± 0.14	$\mathbf{4.97 \pm 0.13}$
	SP	1.93 ± 0.31	0.55 ± 0.14	0.47 ± 0.18	$\mathbf{0.11 \pm 0.01}$
2	HV	2.28 ± 0.07	4.96 ± 0.11	4.10 ± 0.23	$\mathbf{5.37 \pm 0.09}$
	SP	2.25 ± 0.23	0.29 ± 0.04	0.58 ± 0.23	$\mathbf{0.07 \pm 0.01}$

Table 2: Performance w.r.t reward distance metrics: 1 and 2 represents MO-Cheetah and MO-AntXY, respectively.

Table 3 shows the effect of our preference-conditioned diffusion model. ParIRL+BC denotes distillation using the naive BC algorithm. We test ParIRL+DU with varying guidance weights δ in (14), ranging from 0.0 to 1.8. The results indicate that ParIRL+DU improves by 6.42% at average over ParIRL+BC. Employing both unconditioned and conditioned policies ($\delta > 0$) contributes to improved performance.

Env.	Met.	ParIRL+BC	$\delta = 0.0$	$\delta = 1.2$	$\delta = 1.8$
1	HV	4.52 ± 0.46	4.86 ± 0.09	$\mathbf{4.96 \pm 0.06}$	4.94 ± 0.05
	SP	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
2	HV	5.48 ± 0.05	5.54 ± 0.10	5.61 ± 0.10	$\mathbf{5.65 \pm 0.07}$
	SP	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00

Table 3: Performance of preference-conditioned diffusion models: 1 and 2 represent MO-Swimmer and MO-AntXY, respectively.

4.4 Case Study on Autonomous Driving

To verify the applicability of our framework, we conduct a case study with autonomous driving scenarios in the CARLA simulator [Dosovitskiy *et al.*, 2017]. In Figure 5, the comfort mode agent drives slowly without switching lanes, while the sport mode agent accelerates and frequently switches lanes (indicated by dotted arrow) to overtake front vehicles (highlighted by dotted circle) ahead. Using the distinct datasets collected from these two different driving modes, ParIRL generates a set of diverse *custom* driving policies. Specifically, as depicted in the bottom of Figure 5, the closer the custom agent's behavior is to the sport mode, the more it tends to switch lanes (increasing from 0 to 2) and to drive at higher speeds with lower energy efficiency. The agent in custom mode-2 balances between the comfort and sport modes well, maintaining the moderate speed and changing lanes once.

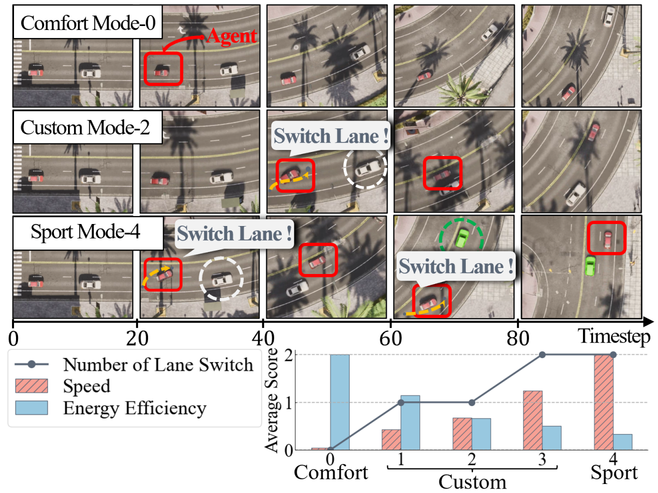


Figure 5: Visualization of agents obtained via ParIRL: the red rectangle denotes learned agent, the dotted lines denote lane changes, the dotted circles denote the vehicles overtaken by our agent.

5 Related Work

Multi-objective RL. In the RL literature, several multi-objective optimization methods were introduced, aiming at providing robust approximation of a Pareto policy set. [Yang *et al.*, 2019; Xu *et al.*, 2020] explored Pareto policy set approximation through reward scalarization in online settings, where reward signals are provided. Recently, [Zhu *et al.*, 2023] proposed the Pareto decision transformer in offline settings, requiring a comprehensive dataset that covers all preferences. These prior works and ours share a similar goal to achieve a tradeoff-aware agent based on Pareto policy set approximation. However, different from the prior works, our work concentrates on practical situations with the strictly limited datasets and without any rewards from the environment.

Inverse RL. To infer a reward function from datasets, IRL has been investigated along with adversarial schemes. [Fu *et al.*, 2018] established the practical implementation of IRL based on the generative adversarial framework; which was further investigated by [Zeng *et al.*, 2022; Garg *et al.*, 2021; Wang *et al.*, 2024]. Recently, [Kishikawa and Arai, 2021] introduced a multi-objective reward function recovery method, using a simple discrete grid-world environment. Contrarily, our ParIRL targets the approximation of a Pareto policy set. Instead of exploring the linear combinations of rewards, ParIRL employs the reward distance metric, and further, optimizes the performance lower bound of learned policies.

Reward Function Evaluation. Reward function evaluation is considered important in the RL literature, but was not fully investigated. [Gleave *et al.*, 2020] first proposed the EPIC by which two reward functions are directly compared without policy optimization, and verified that the policy regret is bounded. This was extended by [Wulfe *et al.*, 2022] for mitigating erroneous reward evaluation. However, those rarely investigated how to use such metrics for multi-objective learning. Our work is the first to conjugate reward function evaluation for Pareto policy set approximation in IRL settings.

6 Conclusion

We presented the ParIRL framework to induce a Pareto policy set from strictly limited datasets in terms of preference diversity. In ParIRL, the recursive IRL with the reward distance regularization is employed to achieve the Pareto policy set. The set is then distilled to the preference-conditioned diffusion policy, enabling robust policy adaptation to unseen preferences and resource efficient deployment. Our framework is different from the existing IRL approaches in that they only allow for imitating an individual policy from given datasets.

Acknowledgements

We would like to thank anonymous reviewers for their valuable comments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-01045, 2022-0-00043, 2021-0-00875, 2020-0-01821, 2019-0-00421) and by the National Research Foundation of Korea (NRF) grant funded by the MSIT (No. RS-2023-00213118) and by Samsung electronics.

References

[Dosovitskiy *et al.*, 2017] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Conference on Robot Learning*, pages 1–16, 2017.

[Fu *et al.*, 2018] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[Garg *et al.*, 2021] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. IQ-Learn: Inverse soft-Q Learning for Imitation. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, pages 4028–4039, 2021.

[Gleave *et al.*, 2020] Adam Gleave, Michael Dennis, Shane Legg, Stuart Russell, and Jan Leike. Quantifying differences in reward functions. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[Ho and Ermon, 2016] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, 2016.

[Ho and Salimans, 2022] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, 2022.

[HO *et al.*, 2020] Jonathan HO, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Proceedings of the 34th Conference on Neural Information Processing System*, 2020.

[Kishikawa and Arai, 2021] Daiko Kishikawa and Sachiyo Arai. Multi-objective inverse reinforcement learning via non-negative matrix factorization. In *10th International Congress on Advanced Applied Informatics*, pages 452–457, 2021.

[Kyriakis *et al.*, 2022] Panagiotis Kyriakis, Jyotirmoy v. Deshmukh, and Paul Bogdan. Pareto policy adaptation. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[Pearce *et al.*, 2023] Tim Pearce, Tabish Rashid, Anssi Kanervisto, David Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, and Sam Devlin. Imitating Human Behaviour with Diffusion Models. In *Proceddings of the 11th Conference on Learning Representations*, 2023.

[Shafiullah *et al.*, 2022] Nur Muhammad Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior Transformers: Cloning k modes with one stone. In *Proceedings of the 37th Conference on Nueral Information Processing Systems*, 2022.

[Villani, 2003] Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Society, 2003.

[Wang *et al.*, 2024] Bingzheng Wang, Guoqiang Wu, Teng Pang, Yan Zhang, and Yilong Yin. DiffAIL: Diffusion Adversarial Imitation Learning. In *Proceedings of the 38th Conference on Artificial Intelligence*, pages 4028–4039, 2024.

[Wulfe *et al.*, 2022] Blake Wulfe, Logan Ellis, Jean Mercat, Rowan McAllister, Adrien Gaidon, and Ashwin Balakrishna. Dynamics-aware comparison of learned reward functions. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[Xu *et al.*, 2020] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10608–10616, 2020.

[Yang *et al.*, 2019] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, pages 14610–14621, 2019.

[Zeng *et al.*, 2022] Siliang Zeng, Chenliang Li, Alfredo Garcia, and Mingyi Hong. Maximum-Likelihood Inverse Reinforcement Learning with Finite-Time Guarantees. In *Proceedings of the 37th Conference on Nueral Information Processing Systems*, 2022.

[Zhu *et al.*, 2023] PBaiting Zhu, Meihua Dang, and Aditya Grover. Scaling Pareto-Efficient Decision Making Via Offline Multi-Objective RL. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.