# LEARNING FROM DIVERSE EXPERTS: BEHAVIOR ALIGNMENT THROUGH MULTI-OBJECTIVE INVERSE REINFORCEMENT LEARNING

**Jun-Jie Yang**[1*] **Qian-You Zhang**[1*] **Chia-Heng Hsu**[1] **Xi Liu**[2] **Ping-Chun Hsieh**[1]
[1]National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[2]Applied Machine Learning, Meta AI, Menlo Park, CA, USA
{yang9187.cs09, jxea666.cs11, pinghsieh}@nycu.edu.tw

## ABSTRACT

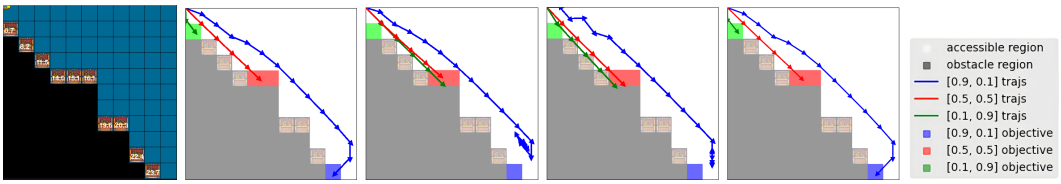
Imitation learning (IL) from demonstrations serves as one data-efficient and practical framework for achieving human-level performance and behavior alignment with human experts in sequential decision making. However, existing IL approaches mostly presume that the expert demonstrations are homogeneous and largely ignore the practical issue of multiple performance criteria and the resulting diverse preferences of the experts. To tackle this, we propose to learn simultaneously from multiple experts of different preferences through the lens of multi-objective inverse reinforcement learning (MOIRL). Specifically, MOIRL achieves unified learning from diverse experts by inferring the vector-valued reward function of each expert and reconcile these via *reward consensus*. Built on this, we propose Multi-Objective Inverse Soft-Q Learning (MOIQ), which penalizes differences in the rewards for encouraging reward consensus. This approach enjoys transferability to unseen preferences due to the reward consensus among demonstrators. To further annotate the unknown preferences of demonstrations, we introduce a posterior network that can predict preferences of the given trajectories. Extensive experiments demonstrate that MOIQ is competitive in challenging scenarios with low and noisy annotations and can outperform stronger benchmark methods and approaches expert-level performance in the fully annotated regime.

## 1 INTRODUCTION

Imitation learning (IL) from expert demonstrations serves as a data-efficient and practical framework for achieving behavior alignment with the experts as well as human-level performance in sequential decision making, especially for those real-world applications where domain expertise is available for warm start or reward signal is sparse or difficult to design. For example, in robot control (Finn et al., 2016), by leveraging demonstrations from humans, the reward signal becomes implicitly embedded in the observed expert behaviors, enabling the robot to learn complex tasks without explicitly defined reward functions. As a result, this alleviates the challenges associated with manually crafting reward structures, which can be intricate and often elusive in capturing nuanced task requirements. In addition to robot control, for similar reasons, various other real-world applications also benefit significantly from demonstrations for behavior alignment, such as autonomous driving (Le Mero et al., 2022; Codevilla et al., 2018), financial trading (Liu et al., 2020), recommender systems (Chen et al., 2021; 2023c), and medical treatments (Wang et al., 2020; 2022).

Existing IL approaches mostly presume that the expert demonstrations are *homogeneous* in the sense that they all reflect the same or similar expert behavior. However, this presumption typically does not hold in practice, at least for the following two reasons. First, for ease of deployment, the expert demonstrations are usually collected from multiple demonstrators. Second, a plethora of real-world sequential decision-making problems inherently involve the joint optimization of multiple performance criteria, some of which could even be conflicting. As a result, the *preference over multiple performance criteria* is naturally and implicitly encoded into the expert behavior, and therefore the

[1*]Equal contribution.




(a) DST (b) Expert (c) InfoGAIL (d) Ess-InfoGAIL (e) MOIQ

Figure 1: (a) DST environment. The expert with a higher preference for valuable treasures will seek out more distant treasures, whereas the one with a higher preference for minimizing step costs will tend to stay closer. (b) The demonstrations of experts are imbalanced in total steps, where green trajectory took 2 steps to reach its treasure, red trajectory took 5, and blue trajectory took 13. (c)(d) InfoGAIL and Ess-InfoGAIL struggle to distinguish 3 experts and tend to overlook the shorter (green) demonstrations. They both fail to recognize that the furthest (blue) trajectory requires a turn at the end. (e) MOIQ perfectly mimics experts with three different targets. Notably, the furthest trajectory accurately making a turn demonstrates the advantage of IRL in recovering a reward, indicating that our approach can retrieve an accurate reward from multi-expert demonstrations.

preferred behavior can be rather diverse and shall vary with the preference. For example, robot locomotion tasks involve trade-offs between speed and energy use. The desired behavior vary with the user's preference (*e.g.,* prioritizing speed or battery life). The performance of navigation tasks includes reaching goals, time efficiency, and energy usage. Therefore, the desired route naturally depends on the preference over these objectives of interest.

Throughout this paper, we refer to this problem setting as *Learning from Diverse Experts* (LfDE). Among the vast IL literature, there are very few prior works on addressing LfDE-related formulations, and the most relevant ones are built on the idea of *imitation learning with latent contexts*, *e.g.,* obtained through maximizing mutual information like in InfoGAIL-like methods (Li et al., 2017; Fu et al., 2023) or via latent skills with expertise level estimation (Beliaev et al., 2022). However, latent contexts do not have a clear semantic interpretation and are oblivious to the multi-objective and preference-dependent structure. As shown in Figure 1, InfoGAIL and ESS-InfoGAIL struggle to distinguish among three experts and tend to overlook the shorter demonstrations in the Deep Sea Treasure environment. Therefore, this motivates one important and open research question: *How to train an imitator model that learns jointly from diverse demonstrators over multiple performance criteria and can adapt well to a wide range of preferences at deployment?*

To tackle this challenge, we propose a holistic framework to solve LfDE through the lens of Multi-Objective Inverse Reinforcement Learning (MOIRL). Unlike traditional Single-Objective IRL where a single reward function is inferred, MOIRL infers a vector-valued reward function with one dimension for each performance objective. A key challenge is that these inferred rewards can vary significantly between demonstrators. To address this, we introduce the concept of *reward consensus*, enforcing the inferred rewards to converge through a consensus constraint. For discrete environments, we reformulate MOIRL as a global variable consensus problem and solve it using the ADMM method (Boyd et al., 2011), suitable for tasks with small state and action spaces. For continuous environments, we propose MOIQ, which converts the consensus constraint into a penalty term embedded in the soft Q-function (Garg et al., 2021), enabling efficient optimization using off-the-shelf deep learning frameworks.

**Contributions.** We summarize our contributions as follows: (1) We propose the MOIRL framework, which learns jointly from demonstrations of diverse preferences over multiple objectives through reward consensus and achieves knowledge sharing across preferences and effective transfer to unseen preferences during training. (2) We present MOIQ, which serves as a practical implementation of MOIRL and can learn to imitate the experts of diverse preferences from demonstrations with very little and possibly noisy knowledge about the preference. (3) Through extensive experiments, we demonstrate that the proposed MOIRL outperforms the benchmark IRL methods and can learn efficiently in the full-annotation, low-annotation, and noisy-annotation settings, as well as in complex environments where other baselines fail to train effectively.

## 2 PRELIMINARIES

In this section, we describe the background information needed for the proposed MOIRL framework, including multi-objective MDPs and the standard inverse RL formulation.

**Multi-Objective Markov Decision Process (MOMDP):** An MOMDP can be fully characterized by the tuple $(\mathcal{S}, \mathcal{A}, p_0, \mathcal{P}, \boldsymbol{r}, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $p_0 \in \Delta(\mathcal{S})$[1] is the initial state distribution, $\mathcal{P}$: $\mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function, $\boldsymbol{r}$: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is the vector-valued reward function with $d$ denoting the number of objectives, and $\gamma \in (0, 1)$ is the discount factor. Let $\Pi := \{\pi | \pi : \mathcal{S} \to \Delta(\mathcal{A})\}$ denote the set of all Markovian policies (possibly randomized). Let $\mathcal{R}$ and $\Omega$ denote the set of possible reward functions and the set of all preferences, respectively. For a policy $\pi \in \Pi$, the occupancy measure $\rho_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ (also known as the discounted state-action distribution) is defined as $\rho_\pi(s, a) := (1 - \gamma)\pi(a|s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$. Notably, when $d = 1$, the above reduces to the standard formulation of single-objective MDPs.

In this paper, we focus on the standard linear scalarization function in the MOMDP literature (Abels et al., 2019; Yang et al., 2019; Hung et al., 2023; Basaklar et al., 2023), *i.e.,* the vectorized rewards can be scalarized by an affine function $\mathfrak{S}_{\boldsymbol{\omega}} : \mathbb{R}^d \to \mathbb{R}$ with a preference vector $\boldsymbol{\omega} \in \Omega$ such that

$$\mathfrak{S}_{\boldsymbol{\omega}}(\boldsymbol{r}(s, a)) := \boldsymbol{\omega}^\top \boldsymbol{r}(s, a) \equiv r_{\boldsymbol{\omega}}(s, a) \tag{1}$$

where $\boldsymbol{\omega}$ is a $d$-dimensional preference vector and $r_{\boldsymbol{\omega}}$ serves as a shorthand for the scalarized reward function under $\boldsymbol{\omega}$. Under linear scalarization, without loss of generality, we presume that the preference vectors lie in a unit simplex for simplicity.

**Inverse RL.** As one major paradigm of imitation learning, inverse RL (IRL) imitates the expert behavior by first inferring the underlying reward function that aligns with the expert demonstrations and thereafter employs an off-the-shelf RL algorithm to obtain the corresponding optimal policy (Ng & Russell, 2000; Abbeel & Ng, 2004). Given that IRL is known to be an underdetermined problem in the sense that the expert policy can be optimal under multiple reward functions (Osa et al., 2018), maximum causal entropy IRL (Ziebart et al., 2008; 2010; Ho & Ermon, 2016) addresses this identification issue by reformulating IRL as the following optimization problem. Let $\rho_e$ denote the occupancy measure of the expert policy to be imitated. Given a class of candidate scalar reward functions $\mathcal{R}$, we jointly solve for the unknown reward function and the imitation policy as

$$\max_{r \in \mathcal{R}} \Big( \min_{\pi \in \Pi} -\mathbb{E}_{\rho_\pi}[r(s, a)] - H(\pi) \Big) + \mathbb{E}_{\rho_e}[r(s, a)] - \psi(r), \tag{2}$$

where $H(\pi) := \mathbb{E}_{(s,a)\sim\rho_\pi}[-\log \pi(a|s)]/(1 - \gamma)$ is the discounted causal entropy of a policy $\pi$ and $\psi : \mathbb{R}^{\mathcal{S} \times \mathcal{A}} \to \mathbb{R}$ is a convex regularizer. Notably, the maximin problem in Equation (2) can be viewed as the dual problem of matching the occupancy measures between the expert and the imitator from the perspective of optimal transport (Xiao et al., 2019). To solve the problem in Equation (2), one natural approach is to alternate between reward inference and policy learning via RL, which could incur substantial computational cost. To address this, (Ho & Ermon, 2016) pinpoints the connection between IRL and the generative adversarial network (GAN) (Goodfellow et al., 2014) by showing the equivalence between Equation (2) under a proper regularizer and the minimax problem for training GANs.

**Soft $Q$-Function and Inverse Soft Bellman Operator:** Despite the efficacy of Equation (2), the corresponding adversarial training can be rather unstable in practice. To obviate the need for adversarial training, (Garg et al., 2021) proposed to further characterize the relation between the reward and the $Q$ function.

For any given reward function $r$ and any policy $\pi$, define the soft Bellman operator $\mathcal{B}_r^\pi : \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ as $(\mathcal{B}_r^\pi Q)(s, a) = r(s, a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^\pi(s')]$, where $V^\pi(s) := \mathbb{E}_{a\sim\pi(\cdot|s)}[Q(s, a) - \log \pi(a|s)]$. Notably, this operator uniquely characterizes the *soft Q-function*, which is defined as the solution to the soft Bellman equation as $Q = \mathcal{B}_r^\pi Q$ (Geist et al., 2019). Built on this, (Garg et al., 2021) introduces the *inverse soft Bellman operator* $\mathcal{T}^\pi$ defined as

$$(\mathcal{T}^\pi Q)(s, a) := Q(s, a) - \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^\pi(s')]. \tag{3}$$

Through the operator $\mathcal{T}^\pi$, one can view $\mathcal{T}^\pi Q$ as a $Q$-induced reward function. In fact, it has been established that the soft $Q$-function and the $Q$-induced reward function enjoy a bijection under

[1]Throughout this paper, for a set $\mathcal{X}$, we use $\Delta(\mathcal{X})$ to denote the set of all probability distributions over $\mathcal{X}$.

$\mathcal{T}^\pi$ (Garg et al., 2021). By leveraging inverse soft Bellman operator and an appropriate definition of reward regularizer $\psi$, the maximin problem in Equation (2) can be further converted into an alternative minimax problem in the $Q$-policy space with the objective function as

$$\mathcal{J}(\pi, Q) = \mathbb{E}_{\rho_e}\Big[\phi\big(Q(s,a) - \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[V^\pi(s')]\big)\Big] - \underbrace{(1-\gamma)\mathbb{E}_{p_0}[V^\pi(s_0)]}_{V_0 \text{ loss}} \tag{4}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is a concave function and $p_0$ is the initial state distribution. This optimization problem can be solved by a standard actor-critic RL algorithm, such as Soft Actor-Critic (Haarnoja et al., 2018), and thereby obviates the need for adversarial training.

## 3 Problem Statement

Before providing the formal problem formulation, we first present an illustrative example to further explain and clarify the relationship between objective and preference. Inherent to any environment is a set of performance criteria, which encompasses multiple conceivable long-term metrics or goals. For example: (i) In robot locomotion tasks (Aller et al., 2019), the major performance criteria typically include energy efficiency, moving speed, robustness to external disturbance, and human likeness in locomotion behavior. (ii) In autonomous driving, the common long-term performance criteria typically include safety metrics, such as scene drivability and collision-based risks (Guo et al., 2019), as well as driving stability and fuel efficiency. Notably, despite that these long-term criteria are pre-determined, this does not imply that the reward signal can be easily designed accordingly. Based on the above motivation, we now present our problem statement.

**Learning from Diverse Experts:** Let $\mathcal{D} = \{(s_1^{(i)}, a_1^{(i)}, \cdots, s_T^{(i)}, a_T^{(i)})\}$ denote a collection of demonstration trajectories from multiple diverse experts where $s_t^{(i)}$ and $a_t^{(i)}$ denote the state and action of the $i$-th trajectory at time step $t$. There exist $d$ common performance criteria that are shared among the set of experts, and each expert performs the demonstrations based on its individual preference over these performance criteria. More specifically, we assume that there is an oracle capable of assigning a preference label $\omega \in \Omega$ to each trajectory with respect to the performance criteria. There are two major settings considered in this paper: (i) *Low-annotation regime*: Only a small part of the trajectory labels are known to the learner. (ii) *Full-annotation regime*: All the trajectory labels are provided to the learner. The goal is to learn a conditional policy $\pi(a|s, \omega)$ that can align the behavior of the learner with multiple experts under diverse preferences.

**Remark on the Annotation of Preference Labels.** While it is intuitively desirable to directly learn from diverse demonstrations without any preference annotation, such unsupervised learning from heterogeneous and unstructured data is known to be challenging, especially in the case of behavior alignment. Specifically, recent research attempts have uncovered that unsupervised learning of disentangled representations is inherently infeasible without any prior inductive biases on either the model or the dataset (Locatello et al., 2019; Fu et al., 2023). For example, in the DST environment (see Figure 1a), the performance criteria are treasure value and step cost. The trajectory that reaches the farthest treasure has a preference of [0.9, 0.1]. In this case, the preference that can explain the same behavior ranges from [0.607, 0.393] to [1, 0]. This shows that there can be multiple ambiguous preferences, *i.e.,* a unique preference does not necessarily exist in the unsupervised setting.

Given this, we first assume that the preferences are known in Section 4.1 for didactic purposes. In Section 4.2, we will further discuss how our method works in the low-annotation regime.

## 4 Methodology

### 4.1 MOIRL Framework With Reward Consensus

In this section, we formally introduce the MOIRL framework, which serves as a unified approach to learning from diverse experts. The proposed MOIRL is built on the principle of *reward consensus*, substantiated in two steps: (i) We extend the maximin problem in Equation (2) to accommodate the demonstrations from $n$ different experts with the help of vector-valued rewards as in MOMDP. (ii) We propose an additional *reward consensus constraint* to enforce that the vector-valued rewards in-

ferred by different experts are as consistent as possible. Specifically, this framework can be formally stated as follows[2]

$$\max_{\{\boldsymbol{r}_i\}\in\mathcal{R}^n}\min_{\{\pi_i\}\in\Pi^n}\mathcal{J}\big(\{\pi_i\}_{i=1}^n,\{\boldsymbol{r}_i\}_{i=1}^n,\{\boldsymbol{\omega}_i\}_{i=1}^n\big):=\sum_{i=1}^n\omega_i^\top(\mathbb{E}_{\rho_{e_i}}[\boldsymbol{r}_i(s,a)]-\mathbb{E}_{\rho_i}[\boldsymbol{r}_i(s,a)])$$
$$\text{subject to}\quad \boldsymbol{r}_1=\boldsymbol{r}_2=\cdots=\boldsymbol{r}_n$$

where $\boldsymbol{\omega}_i$ is the preference of the $i$-th expert. Notably, such optimization problem is known to be a *global consensus problem*, which can be efficiently solved by employing alternating direction method of multipliers (ADMM) (Boyd et al., 2011). Specifically, given the initial $\pi_1^0$, ..., $\pi_n^0$, ADMM solves this reward consensus problem by iteratively update the variables as follows: for each $i$,

$$\begin{aligned}\boldsymbol{r}_i^{k+1}&=\arg\max_{\boldsymbol{r}_i\in\mathcal{R}}\ \boldsymbol{\omega}_i^\top\big(\mathbb{E}_{\rho_{e_i}}[\boldsymbol{r}_i(s,a)]-\mathbb{E}_{\rho_i}[\boldsymbol{r}_i(s,a)]\big)-(\rho/2)||\boldsymbol{r}_i-\bar{\boldsymbol{r}}^k+\boldsymbol{u}_i^k||_2^2,\\ \boldsymbol{u}_i^{k+1}&=\boldsymbol{u}_i^k+\boldsymbol{r}_i^{k+1}-\bar{\boldsymbol{r}}^{k+1},\end{aligned}\tag{5}$$

where $\rho>0$ is the weight of quadratic penalty in ADMM and $\bar{\boldsymbol{r}}^k:=\frac{1}{n}\sum_{i=1}^n\boldsymbol{r}_i^k$ denotes the inferred reward averaged over experts. With the inferred common reward, we can train $n$ agents by running an off-the-shelf RL algorithm to obtain $\{\pi_i^j\}_{i=1}^n$ accordingly. By repeating this procedure for sufficiently many rounds, the inferred reward is expected to converge to the true reward.

**Motivating Experiments:** For didactic purposes, we first evaluate our algorithm on discrete DST with two-dimensional rewards. We learn from three experts with preferences [0.9, 0.1], [0.5, 0.5], and [0.1, 0.9]. As depicted in Figure 2, all agents reach near-optimal reward within 10 rounds. This showcases the performance of our approach in the DST environment, indicating the idea of learning a common reward function among agents indeed helps.

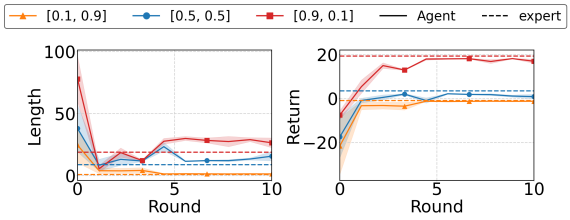



Figure 2: **Comparison of MOIQ and the expert.** The results are presented in terms of return and trajectory length averaged over 5 random seeds. A *Round* is defined as the completion of one iteration incorporating the MOIRL algorithm with consensus ADMM and the RL algorithm.

### 4.2 Practical Implementation of MOIRL for Continuous Control

**Multi-Objective Inverse Soft-Q Learning.** To implement MOIRL for the practical continuous control, we extend the concept of reward consensus to the minimax problem in Equation (4). As the reward consensus constraint would result in a constrained RL problem, we simplify the training by introducing a penalty term, which is the $\ell_2$ norm between the difference of each reward $\boldsymbol{r}_i$. Notably, by further decomposing the problem into $n$ separate optimization objectives, we can optimize each agent $i$ as follows:

$$\begin{aligned}\mathcal{J}(\pi_i,Q_i,\omega_i)&=\mathbb{E}_{\rho_{e_i}}\Big[\phi\big(\boldsymbol{\omega}_i^\top(\boldsymbol{Q}_i(s,a)-\gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\boldsymbol{V}^{\pi_i}(s'))\big)\Big]\\ &-\mathbb{E}_\mu\big[\boldsymbol{\omega}_i^\top(\boldsymbol{V}^\pi(s)-\gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\boldsymbol{V}^\pi(s'))\big]-\psi(\boldsymbol{\omega}_i^\top\boldsymbol{r}_i)-\beta\sum_{j=1}^n\|\boldsymbol{r}_i-\boldsymbol{r}_{j+1}\|_2.\end{aligned}\tag{6}$$

The regularizer $\psi$ is chosen based on the environment, with details provided in Appendix A.7.

**Preference annotation.** To determine which preference a given trajectory belongs to, we introduce a posterior network to predict the preference. The idea is to maximize the mutual information between the preference and the trajectory. Unlike (Li et al., 2017), which simplifies the problem by representing the trajectory with a single state-action pair, our approach leverages full trajectories of arbitrary length, ensuring a more reliable preference estimation.

[2]For ease of exposition, we ignore the entropy and the reward regularizer in this subsection for brevity. That being said, our framework is readily capable of accommodating these regularization terms.

Due to the existence of multiple ambiguous preferences within the MOIRL framework, we employ a small portion of preference-annotated trajectories to facilitate the learning of accurate preferences (Fu et al., 2023). We sample trajectories from both experts and agents to train the posterior network:

$$\mathcal{L}(\hat{P}) = \mathbb{E}_{\omega\sim p(\omega),\tau\sim\rho_\pi^\omega}[\log \hat{P}(\omega|\tau)] + \mathbb{E}_{\omega\sim p(\omega),\tau\sim\rho_e^\omega}[\log \hat{P}(\omega|\tau)] + 2H(\omega) \tag{7}$$

where $\hat{P}$ is the posterior network, $H(\omega)$ is the entropy term, $\rho_\pi^\omega$ and $\rho_E^\omega$ represent the distributions of trajectories conveying the preference $\omega$ from agents and experts, respectively.

We employ an actor-critic framework to learn under different preferences. The actor network optimizes the policy based on the estimated Q-values, while the critic network learns and estimates Q-values among various preference. MOIQ-PA extends MOIQ by adding a posterior network $\hat{P}_\theta$ to capture expert preferences. Practical algorithms and network updates are detailed in Appendix A.1

# 5 Experiments

## 5.1 Experimental Setup

**Diverse Preferences of Expert Demonstrations:** We train our experts with various preferences: For 2-dimensional environments (DST, Mo-Halfcheetah, Mo-Walker, Mo-Ant) using [0.9, 0.1], [0.5, 0.5], [0.1, 0.9], for the 3-dimensional environment (Mo-Hopper) using [0.8, 0.1, 0.1], [0.1, 0.8, 0.1], [0.1, 0.1, 0.8], and for the 5-dimensional environment (Mo-Humanoid) using [0.6, 0.1, 0.1, 0.1, 0.1], [0.1, 0.6, 0.1, 0.1, 0.1], [0.1, 0.1, 0.6, 0.1, 0.1], [0.1, 0.1, 0.1, 0.6, 0.1], [0.1, 0.1, 0.1, 0.1, 0.6].

**Environments:** For Mo-HalfCheetah, Mo-Hopper, we directly use MO-Gym (Felten et al., 2023), which is a multi-objective gymnasium environment. For DST, we modify both the state and action space of discrete DST from MO-Gym to a 2-dimensional continuous space. For Mo-Walker and Mo-Ant, we inherit the classes of Walker2d and Ant from Gymnasium (Towers et al., 2024) and extend the reward space to two dimensions. Similarly, for Mo-Humanoid, we extend the Humanoid class to a five-dimensional reward space with a 378-dimensional observation space and a 17-dimensional action space. It is important to note that the Humanoid environment in MuJoCo is one of the more complex environments. In the environments such as Mo-HalfCheetah, Mo-Walker, and Mo-Humanoid, where expert performance is more challenging, we use 1M training steps. For other environments, we limit the training steps to 0.5M. Descriptions of environments, rewards, and more details are provided in the appendix A.2.

**Baselines:** In this section, we use these baselines to demonstrate the ability to learn from multi-expert demonstrations. (i) *InfoGAIL:* The original InfoGAIL with a fixed uniform categorical distribution. (ii) *Ess-InfoGAIL:* The original Ess-InfoGAIL without giving any inducement in the form of training reward from environments. (iii) *GAIL:* GAIL with SAC as the generator. (iv) *IQ:* The original IQ. (v) *MOIQ:* The original MOIQ. (vi) *MOIQ-PA:* MOIQ with preference annotation.

## 5.2 Experimental Results and Discussions

In this section, we address four key questions through comprehensive experiments. First, we explore the performance of MOIQ-PA in low-annotation regimes (Q1), presenting a detailed comparison with existing methods. Next, we investigate its transferability to unseen preferences (Q2) and delve deeper into the impact of annotation quality (Q3). Finally, we assess whether our method outperforms stronger baselines while maintaining sample efficiency and achieving expert-level performance in full-annotation settings (Q4). Additional experiments are included in the appendix A.4.

**Evaluation metrics:** We introduce Mutual Information (NMI) to measure the correlation between two clusterings, ranging from 0 to 1, with higher values indicating stronger correlation, and Average Entropy (ENT) to evaluate classification consistency under fixed preferences and latent preference consistency under fixed classifications. For returns, we adopt the normalization method from (Agarwal et al., 2021), where a score of 1 represents expert performance and 0 corresponds to a random policy, allowing comparisons between environments.

**Q1: Can MOIQ-PA perform well in the low-annotation regime?** To evaluate this, we collected 10 expert demonstrations for each distinct preference, with each consisting of *only one trajectory*

labeled with ground-truth preference. As shown in Table 1, MOIQ-PA exhibits exceptional disentanglement ability across environments, outperforming both InfoGAIL and Ess-InfoGAIL in most environments. While InfoGAIL and Ess-InfoGAIL only excel in the Mo-Hopper environment, MOIQ-PA demonstrates consistent performance across all environments. This performance disparity can be attributed to our posterior network, which is conditioned on the entire trajectory, allowing the model to more effectively extract latent preferences from movement patterns in demonstrations.

Table 1: **Evaluation of behavior disentanglement quality.** We collect 100 trajectories for each preference generated by the trained policy, which will later be classified by our pre-trained classifiers. The results are averaged from 5 seeds. Boldface marks cases where performance is the best.

| Env | MOIQ-PA(Ours) | | InfoGAIL | | Ess-InfoGAIL | |
|---|---|---|---|---|---|---|
| | NMI | ENT | NMI | ENT | NMI | ENT |
| DST | **1.0 ± 0** | **0.0 ± 0** | 0.73 ± 0 | 0.29 ± 0 | 0.78 ± 0 | 0.23 ± 0 |
| Mo-HalfCheetah | **0.48± 0.10** | 0.37 ± 0.05 | 0.34 ± 0.23 | 0.57 ± 0.16 | 0.33 ± 0.28 | **0.34 ± 0.10** |
| Mo-Walker | **0.97 ± 0.01** | **0.02 ± 0.01** | 0.36 ± 0.27 | 0.37 ± 0.07 | 0.38 ± 0.34 | 0.33 ± 0.12 |
| Mo-Ant | **0.74 ± 0.06** | **0.25 ± 0.05** | 0.00 ± 0.01 | 0.51 ± 0.07 | 0 ± 0 | 0.54 ± 0 |
| Mo-Hopper | **1 ± 0** | **0 ± 0** | 0.97 ± 0.04 | 0.02 ± 0.04 | 0.80 ± 0.18 | 0.19 ± 0.19 |
| Mo-Humanoid | **0.06 ± 0.05** | **0.84 ± 0.10** | 0.01 ± 0.01 | 0.93 ± 0.11 | 0.04 ± 0.05 | 0.92 ± 0.11 |

Furthermore, as shown in Table 2, MOIQ-PA significantly outperforms Ess-InfoGAIL, consistently achieving over 80% of expert-level performance. Although MOIQ-PA shows low NMI and high ENT in the Mo-Humanoid environment, it surpasses Ess-InfoGAIL in the reward achieved for each preference, where the latter only performs near random policy levels. Note that InfoGAIL is not included, as it lacks a fixed correspondence between latent codes and demonstrated behavioral preferences, making it difficult to manually identify preferences in most environments. A detailed visualization of the training curve, IQM metric, and performance profiles is in the appendix A.3.

Table 2: **Testing return of the best model.** The results are averaged across 5 seeds, with each tested over 100 demonstrations. The expert scores are averaged over ten demonstrations. Boldface marks cases where performance is the best. Underlined results indicate those that outperform the expert.

| Env | DST | | | Mo-HalfCheetah | | | Mo-Walker | | | Mo-Ant | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Preference | [0.9,0.1] | [0.5,0.5] | [0.1,0.9] | [0.9,0.1] | [0.5,0.5] | [0.1,0.9] | [0.9,0.1] | [0.5,0.5] | [0.1,0.9] | [0.9,0.1] | [0.5,0.5] | [0.1,0.9] |
| MOIQ-PA | **1.0 ± 0** | **1.0 ± 0** | **1.0 ± 0** | **0.74 ± 0.05** | **0.99 ± 0.03** | **0.86± 0.05** | **1.04 ± 0.33** | **0.93 ± 0.04** | **0.87 ± 0.14** | **0.85 ± 0.02** | **0.87 ± 0.03** | **1.01 ± 0.01** |
| Ess-InfoGAIL | −0.38 ± 0 | **1.0 ± 0** | −3.54 ± 5.30 | 0.02 ± 0.02 | 0.01 ± 0.01 | −0.09 ± 0.08 | 0.02 ± 0.01 | 0.00 ± 0.00 | 0.39 ± 0.27 | 0.28 ± 0.01 | 0.01 ± 0.05 | −0.44 ± 0.22 |

| Env | Mo-Hopper | | | Mo-Humanoid | | | | |
|---|---|---|---|---|---|---|---|---|
| Preference | [0.8,0.1,0.1] | [0.1,0.8,0.1] | [0.1,0.1,0.8] | [0.6,0.1,0.1,0.1,0.1] | [0.1,0.6,0.1,0.1,0.1] | [0.1,0.1,0.6,0.1,0.1] | [0.1,0.1,0.1,0.6,0.1] | [0.1,0.1,0.1,0.1,0.6] |
| MOIQ-PA | **0.66 ± 0.34** | **1.02 ± 0.23** | **0.85 ± 0.22** | **1.06 ± 0.05** | **0.78 ± 0.02** | **0.63 ± 0.02** | **0.67 ± 0.02** | **0.74 ± 0.30** |
| Ess-InfoGAIL | 0.32 ± 0.31 | 0.04 ± 0.08 | **1.01 ± 0.03** | −0.01 ± 0.01 | 0.00 ± 0.01 | 0.00 ± 0.01 | 0.00 ± 0.01 | -0.22 ± 0.13 |

**Q2: Does MOIQ-PA enjoy good transfer to other unseen preferences?** As shown in Figure 3, we demonstrate the transferability of our model by visualizing the return in two dimensions for environments with 2-dimensional reward space. Additionally, we calculate their respective **Hypervolumes (HV)** and **Expected Utility Maximization (EUM)** scores, as shown in table 3. HV is a key indicator of a model's ability to explore and dominate a multi-objective reward space, with larger values indicating better performance in covering the objective space. EUM measures how well the learned policy maximizes expected rewards in a given environment.

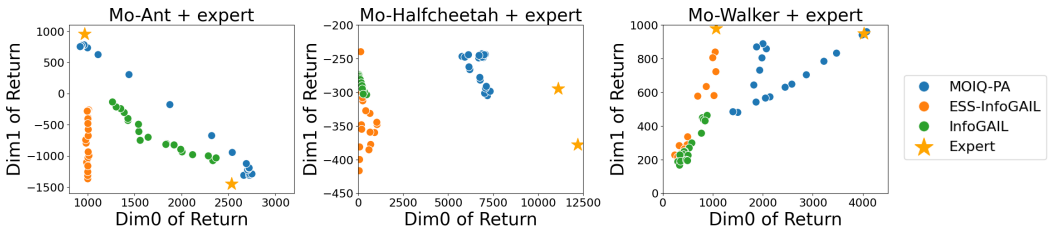



Figure 3: **Transferability of the best-performance model.** Each point is obtained by feeding in a specific preference value from $[1 - 0.05 \times i, 0.05 \times i]$ for $i \in [1, 19]$. Evaluations are conducted over 100 episodes, and the results are averaged across 5 different seeds.

Table 3: **Evaluation of objective space coverage and reward optimization** The HV($\times 10^3$) and EUM results across environments with 2-dimensional reward space. The results are averaged across 5 different seeds. Boldface highlights the best performance.

| | MOIQ-PA (Ours) | | InfoGAIL | | ESS-InfoGAIL | |
|---|---|---|---|---|---|---|
| Env | HV ($\times 10^3$) | EUM | HV ($\times 10^3$) | EUM | HV ($\times 10^3$) | EUM |
| Mo-HalfCheetah | **2896.47 ± 251.35** | **3858.63 ± 337.25** | 146.36 ± 75.04 | 87.40 ± 118.42 | 354.44 ± 174.98 | 401.37 ± 198.91 |
| Mo-Walker | **4042.90 ± 233.37** | **2557.89 ± 90.25** | 455.75 ± 91.75 | 696.58 ± 70.13 | 1012.49 ± 503.75 | 959.20 ± 292.37 |
| Mo-Ant | **8399.82 ± 257.46** | **1291.47 ± 49.80** | 6077.55 ± 365.16 | 878.63 ± 81.69 | 2869.60 ± 216.95 | 417.76 ± 106.13 |

From the results, MOIQ-PA demonstrates strong performance in Mo-HalfCheetah and Mo-Ant, while its results in Mo-Walker are impacted by the absence of intermediate preferences. However, it consistently outperforms InfoGAIL and ESS-InfoGAIL. Notably, MOIQ-PA achieves superior HV and EUM scores across all environments, underscoring its effectiveness in covering the objective space, maximizing rewards, and transferring to unseen preferences.

**Q3: Can MOIQ-PA still perform well in the noisy-annotation regime?** To answer this, we still collected 10 expert demonstrations for each preferences, with each containing *only one trajectory* labeled with *misspecified* preference. As shown in table 4, MOIQ-PA can still have similar performance in most environments under inaccurately labeled demonstrations, demonstrating the robustness of our approach. The exception is the Mo-Hopper environment, where the added noise leads to increased instability, as the environment was already unstable in the original setting.

Table 4: **Comparison between the accurate and inaccurate trials.** MOIQ-PA shows strong performance with inaccurate preferences, deviating minimally from accurate ones. The results are averaged across 3 seeds, with boldface marking cases that perform the best. The preferences listed in the table represent the *inaccurate preferences* for each environment.

| Env | DST | | | Mo-HalfCheetah | | | Mo-Walker | | | Mo-Ant | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Preference | [0.8,0.2] | [0.4,0.6] | [0.2,0.8] | [0.8,0.2] | [0.4,0.6] | [0.2,0.8] | [0.8,0.2] | [0.4,0.6] | [0.2,0.8] | [0.8,0.2] | [0.4,0.6] | [0.2,0.8] |
| Accurate | **1.0 ± 0** | **1.0 ± 0** | **1.0 ± 0** | 0.74 ± 0.05 | **0.99 ± 0.03** | 0.86 ± 0.05 | **1.04 ± 0.33** | **0.93 ± 0.04** | 0.87 ± 0.14 | 0.85 ± 0.02 | **0.87 ± 0.03** | **1.01 ± 0.01** |
| Inaccurate | **1.0 ± 0** | **1.0 ± 0** | **1.0 ± 0** | **0.75 ± 0.04** | 0.97 ± 0.07 | **0.90 ± 0.06** | 1.02 ± 0.09 | 0.79 ± 0.16 | **0.97 ± 0.08** | **0.92 ± 0.04** | **0.87 ± 0.02** | 1.00 ± 0.01 |

| Env | Mo-Hopper | | | Mo-Humanoid | | | | |
|---|---|---|---|---|---|---|---|---|
| Preference | [0.6,0.2,0.2] | [0.3,0.6,0.1] | [0.1,0.3,0.6] | [0.4,0.15,0.15,0.15,0.15] | [0.15,0.4,0.15,0.15,0.15] | [0.15,0.15,0.4,0.15,0.15] | [0.15,0.15,0.15,0.4,0.15] | [0.15,0.15,0.15,0.15,0.4] |
| Accurate | **0.66 ± 0.34** | **1.02 ± 0.23** | **0.85 ± 0.22** | 1.06 ± 0.05 | 0.78 ± 0.02 | 0.63 ± 0.02 | 0.67 ± 0.02 | 0.74 ± 0.30 |
| Inaccurate | 0.59 ± 0.17 | 0.59 ± 0.34 | 0.34 ± 0.30 | **1.08 ± 0.07** | **0.86 ± 0.04** | **0.67 ± 0.02** | **0.69 ± 0.02** | **1.08 ± 0.51** |

**Q4: Do our methods outperform other baselines and achieve expert performance in fully annotation?** We collected 10 expert demonstrations for each preference, with trajectories labeled by ground-truth preferences. We evaluate our method against two baselines: GAIL and IQ-Learn, two single-objective IRL algorithms. These baselines are employed in place of the previously used multi-objective IRL approaches, providing a more accurate comparison. Unlike the baselines, which train separately on each preference and require $n$-times more environment interactions (where $n$ is the number of preferences), MOIQ trains simultaneously on all preferences. This simultaneous training approach highlights the efficiency and effectiveness of MOIQ in leveraging shared knowledge across preferences without additional environment interactions for each preference.

As demonstrated in Table 5, MOIQ consistently outperforms both GAIL and IQ-Learn across most environments and even surpass experts in some. This is because an expert with a particular preference may dominate the performance of another expert under a different preference. In other words, in a single expert setting where each expert is trained independently, agents cannot observe the superior performance of experts with different preferences. This result highlights and underscores the benefits and necessity of learning from multi-expert demonstrations. In particular, in the complex Mo-Humanoid environment, where GAIL and IQ-Learn perform near random levels, MOIQ-PA demonstrates significantly better results, highlighting its capability to handle high-dimensional and complex environments. This demonstrates the advantages of our method in handling complex high-dimensional environments. Overall, our methods exhibit expert-like performance and remarkable sample efficiency. A comprehensive visualization of the IQM metric and performance profiles in the full-annotation regime can be found in in the appendix A.3.

Table 5: **Testing return of the best-performance model.** The results are averaged across 3 random seeds, while the expert scores are averaged over ten demonstrations. Boldface denotes performance within 10% of the expert score, and underline denotes those that outperforms the expert.

| Env | DST | | | Mo-HalfCheetah | | | Mo-Walker | | | Mo-Ant | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Preference | [0.9, 0.1] | [0.5, 0.5] | [0.1, 0.9] | [0.9, 0.1] | [0.5, 0.5] | [0.1, 0.9] | [0.9, 0.1] | [0.5, 0.5] | [0.1, 0.9] | [0.9, 0.1] | [0.5, 0.5] | [0.1, 0.9] |
| MOIQ | **1.0** $\pm$ **0** | **1.0** $\pm$ **0** | **1.0** $\pm$ **0** | **0.94** $\pm$ **0.01** | **1.02** $\pm$ **0.01** | **0.93** $\pm$ **0.02** | **0.99** $\pm$ **0.01** | 0.79 $\pm$ 0.26 | 0.85 $\pm$ 0.05 | **0.99** $\pm$ **0.01** | 0.91 $\pm$ 0.03 | 0.83 $\pm$ 0.01 |
| GAIL | 0.42 $\pm$ 0.58 | **1.0** $\pm$ **0** | 0.92 $\pm$ 0.10 | 0.73 $\pm$ 0.03 | 0.69 $\pm$ 0.04 | 0.59 $\pm$ 0.04 | 0.36 $\pm$ 0.02 | 0.53 $\pm$ 0.04 | **0.96** $\pm$ **0.01** | **1.06** $\pm$ **0.03** | **1.0** $\pm$ **0.01** | 0.87 $\pm$ 0.10 |
| IQ | $-0.07 \pm 0$ | 0.28 $\pm$ 0.02 | **1.0** $\pm$ **0** | 0.01 $\pm$ 0.0 | 0.03 $\pm$ 0.0 | 0.18 $\pm$ 0.0 | **0.99** $\pm$ **0.01** | **0.96** $\pm$ **0.01** | **0.90** $\pm$ **0.01** | **0.89** $\pm$ **0.03** | 0.74 $\pm$ 0.02 | **1.03** $\pm$ **0.0** |

| Env | Mo-Hopper | | | Mo-Humanoid | | | | |
|---|---|---|---|---|---|---|---|---|
| Preference | [0.8, 0.1, 0.1] | [0.1, 0.8, 0.1] | [0.1, 0.1, 0.8] | [0.6, 0.1, 0.1, 0.1, 0.1] | [0.1, 0.6, 0.1, 0.1, 0.1] | [0.1, 0.1, 0.6, 0.1, 0.1] | [0.1, 0.1, 0.1, 0.6, 0.1] | [0.1, 0.1, 0.1, 0.1, 0.6] |
| MOIQ | 0.84 $\pm$ 0.09 | **1.27** $\pm$ **0.10** | **0.97** $\pm$ **0.03** | **1.03** $\pm$ **0.05** | 0.81 $\pm$ 0.02 | 0.59 $\pm$ 0.01 | 0.64 $\pm$ 0.01 | 0.58 $\pm$ 0.19 |
| GAIL | **1.05** $\pm$ **0.07** | **1.38** $\pm$ **0.09** | **1.01** $\pm$ **0.01** | 0.05 $\pm$ 0.01 | 0.06 $\pm$ 0.01 | 0.03 $\pm$ 0.01 | 0.04 $\pm$ 0.01 | 0.06 $\pm$ 0.02 |
| IQ | **0.92** $\pm$ **0.04** | **1.30** $\pm$ **0.04** | **1.01** $\pm$ **0.0** | 0.02 $\pm$ 0.01 | 0.02 $\pm$ 0.01 | 0.05 $\pm$ 0.01 | 0.15 $\pm$ 0.01 | -0.04 $\pm$ 0.01 |

## 6 Related Work

**Imitation learning from single-expert demonstrations:** Addressing the impracticality of solving the max-min optimization problem through nested RL and IRL loops, (Ho & Ermon, 2016) and (Xiao et al., 2019) proposed a framework based on the duality between IRL and occupancy measure, analogous to GANs (Goodfellow et al., 2014). (Fu et al., 2018) presented an adversarial IRL algorithm for robust reward recovery, effective in high-dimensional tasks but inefficient due to its adversarial architecture. Recently, (Garg et al., 2021) proposed a Q-learning approach using an energy-based policy and an inverse soft Bellman operator, simplifying the objective to a single maximization problem over Q space.

**Imitation learning from multi-expert demonstrations.** Recent works have focused on disentangling mixed expert trajectories using latent variables, as seen in extensions of GAIL (Li et al., 2017; Hausman et al., 2017). (Fu et al., 2023) further extended InfoGAIL with semi-supervised GANs to separate behavior representations from imbalanced demonstrations. Another extension, (Kuefler & Kochenderfer, 2018) enhances policy consistency with a learned inference model conditioned on "burn-in" expert demonstrations, maximizing mutual information to avoid degenerate solutions and produce realistic driver models. In addition, (Beliaev et al., 2022) treats demonstrations heterogeneously and considers the demonstrator's expertise using state and demonstrator embeddings. However, these approaches are limited by IL's reliance on the quality and quantity of expert data. (Kishikawa & Arai, 2021) introduced Non-Negative Matrix Factorization (Lee & Seung, 2000) into MOIRL for reward estimation and later incorporated neural networks (Kishikawa & Arai, 2022), but limited to discrete environments, as it necessitates running single-objective IRL. (Chen et al., 2020) employed network distillation to share knowledge from individual strategies. However, the approach struggles with lifelong learning and requires training all components simultaneously. In response, (Chen et al., 2023b) proposed modeling new demonstrations as combinations of prior prototypes, although the methods remain computationally expensive due to repeated IRL calculations.

For a detailed comparison of our method with related works, along with additional discussions on Inverse Constrained Reinforcement Learning and Multi-task Inverse RL, please refer to Appendix A.6.

## 7 Conclusion and Limitations

The limitation of our method is the small portion of annotations needed, which leads to certain requirements for retrieving demonstrations. However, we clearly show that multiple feasible preferences exist within the problem framework, which makes finding the unique ground-truth preference in an unsupervised manner impossible.

We have witnessed the need to consider multiple heterogeneous experts in IRL. Enlightened by this, we utilized the reward consensus among agents. We first conduct a simple and meaningful experiment on a discrete environment to demonstrate that the idea works. We propose MOIQ and its extended version, MOIQ-PA, which can learn the policy under various behaviors of preferences and infer the latent preferences. Various experiments are conducted on both navigation and robotic locomotion tasks, showcasing the capabilities of our method across various aspects, including transferability, disentanglement ability, and, most importantly, the ability to learn from expert demonstrations of diversified preferences.

## ACKNOWLEDGMENTS

This material is based upon work partially supported by National Science and Technology Council (NSTC), Taiwan under Contract No. NSTC 113-2628-E-A49-026, and the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and Ministry of Education, Taiwan. We also thank the National Center for High-performance Computing (NCHC) for providing computational and storage resources.

## REFERENCES

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, 2004.

Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *International conference on Machine Learning*, 2019.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021.

Felix Aller, David Pinto-Fernandez, Diego Torricelli, Jose Luis Pons, and Katja Mombaur. From the state of the art of assessment metrics toward novel concepts for humanoid robot locomotion benchmarking. *IEEE Robotics and Automation Letters*, 5(2):914–920, 2019.

Mattijs Baert, Pietro Mazzaglia, Sam Leroux, and Pieter Simoens. Maximum causal entropy inverse constrained reinforcement learning. *arXiv preprint arXiv:2305.02857*, 2023.

Toygun Basaklar, Suat Gumussoy, and Umit Ogras. PD-MORL: Preference-driven multi-objective reinforcement learning algorithm. In *International Conference on Learning Representations*, 2023.

Mark Beliaev, Andy Shih, Stefano Ermon, Dorsa Sadigh, and Ramtin Pedarsani. Imitation learning by estimating expertise of demonstrators. In *International Conference on Machine Learning*, 2022.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Jiayu Chen, Dipesh Tamboli, Tian Lan, and Vaneet Aggarwal. Multi-task hierarchical adversarial inverse reinforcement learning. In *International Conference on Machine Learning*, 2023a.

Letian Chen, Rohan Paleja, Muyleng Ghuy, and Matthew Gombolay. Joint goal and strategy inference across heterogeneous demonstrators via reward network distillation. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020.

Letian Chen, Sravan Jayanthi, Rohan R Paleja, Daniel Martin, Viacheslav Zakharov, and Matthew Gombolay. Fast lifelong adaptive inverse reinforcement learning from demonstrations. In *Conference on Robot Learning*, 2023b.

Xiaocong Chen, Lina Yao, Aixin Sun, Xianzhi Wang, Xiwei Xu, and Liming Zhu. Generative inverse deep reinforcement learning for online recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.

Xiaocong Chen, Lina Yao, Xianzhi Wang, Aixin Sun, and Quan Z Sheng. Generative adversarial reward learning for generalized behavior tendency inference. *IEEE Transactions on Knowledge and Data Engineering*, 23(10):9878–9889, 2023c.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *IEEE International Conference on Robotics and Automation*, 2018.

Christos Dimitrakakis and Constantin A Rothkopf. Bayesian multitask inverse reinforcement learning. In *Recent Advances in Reinforcement Learning: 9th European Workshop*, 2012.

Florian Felten, Lucas N Alegre, Ann Nowe, Ana Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno C da Silva. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*, 2016.

Huiqiao Fu, Kaiqiang Tang, Yuanyang Lu, Yiming Qi, Guizhou Deng, Flood Sung, and Chunlin Chen. Ess-InfoGAIL: Semi-supervised imitation learning from imbalanced demonstrations. In *Advances in Neural Information Processing Systems*, 2023.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. IQ-Learn: Inverse soft-Q learning for imitation. In *Advances in Neural Information Processing Systems*, 2021.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, 2019.

Adam Gleave and Oliver Habryka. Multi-task maximum entropy inverse reinforcement learning. *arXiv preprint arXiv:1805.08882*, 2018.

Adam Gleave, Mohammad Taufeeque, Juan Rocamonde, Erik Jenner, Steven H Wang, Sam Toyer, Maximilian Ernestus, Nora Belrose, Scott Emmons, and Stuart Russell. imitation: Clean imitation learning implementations. *arXiv preprint arXiv:2211.11972*, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

Junyao Guo, Unmesh Kurup, and Mohak Shah. Is it safe to drive? An overview of factors, challenges, and datasets for driveability assessment in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 21(8):3135–3151, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

Karol Hausman, Yevgen Chebotar, Stefan Schaal, Gaurav Sukhatme, and Joseph J Lim. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 2016.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and JoÃĢo GM AraÃšjo. CleanRL: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

Wei Hung, Bo Kai Huang, Ping-Chun Hsieh, and Xi Liu. Q-Pensieve: Boosting sample efficiency of multi-objective RL through memory sharing of Q-snapshots. In *International Conference on Learning Representations*, 2023.

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

Daiko Kishikawa and Sachiyo Arai. Multi-objective deep inverse reinforcement learning through direct weights and rewards estimation. In *2022 61st Annual Conference of the Society of Instrument and Control Engineers*, 2022.

Alex Kuefler and Mykel J Kochenderfer. Burn-in demonstrations for multi-modal imitation learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018.

Luc Le Mero, Dewei Yi, Mehrdad Dianati, and Alexandros Mouzakitis. A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14128–14147, 2022.

Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, 2000.

Yunzhu Li, Jiaming Song, and Stefano Ermon. InfoGAIL: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*, 2017.

Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. Adaptive Quantitative Trading: An imitative deep reinforcement learning approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2019.

Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, 2000.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2): 1–179, 2018.

Dimitris Papadimitriou, Usman Anwar, and Daniel S Brown. Bayesian methods for constraint inference in reinforcement learning. In *Transactions on Machine Learning Research*, 2022.

Guanren Qiao, Guiliang Liu, Pascal Poupart, and Zhiqiang Xu. Multi-modal inverse constrained reinforcement learning from a mixture of demonstrations. In *Advances in Neural Information Processing Systems*, 2024.

Fumihiro Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2020.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Lu Wang, Wenchao Yu, Xiaofeng He, Wei Cheng, Martin Renqiang Ren, Wei Wang, Bo Zong, Haifeng Chen, and Hongyuan Zha. Adversarial cooperative imitation learning for dynamic treatment regimes. In *Proceedings of The Web Conference*, 2020.

Lu Wang, Ruiming Tang, Xiaofeng He, and Xiuqiang He. Hierarchical imitation learning via subgoal representation learning for dynamic treatment recommendation. In *Proceedings of ACM International Conference on Web Search and Data Mining*, 2022.

Huang Xiao, Michael Herman, Joerg Wagner, Sebastian Ziesche, Jalal Etesami, and Thai Hong Linh. Wasserstein adversarial imitation learning. *arXiv preprint arXiv:1906.08113*, 2019.

Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Advances in Neural Information Processing Systems*, 2019.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2008.

Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. Modeling interaction via the principle of maximum causal entropy. In *International Conference on Machine Learning*, 2010.

# A APPENDIX

## A.1 ALGORITHMS AND NETWORK UPDATE

We present our proposed Algorthim 1 and Algorthim 2 as follows. MOIQ learns from multi-expert demonstrations by updating two networks: the critic network $Q_\phi$ and the actor network $\pi_\psi$. MOIQ-PA extends MOIQ by adding a posterior network $\hat{P}_\theta$ to capture expert preferences. These annotations help guide the learning process and improve performance.

**Algorithm 1** Multi-Objective Inverse soft-Q Learning (MOIQ)

Initialize networks $Q_\phi$ and $\pi_\psi$
**Input:** $\mathcal{D}_{LE}$
**while** environment step $t \leq$ N **do**
  **for** each expert $i$ **do**
    **for** each episode step in [1, T] **do**
      $a_t \sim \pi(\cdot|s_t, \omega_i)$
      $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$
      $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{(s_t, a_t, s_{t+1})\}$
      Update $Q_\phi$ according to Equation (8)
      $\phi_{t+1} \leftarrow \phi_t + \lambda_Q \nabla_\phi \mathcal{J}(Q, i)$
      Update $\pi_\psi$ according to Equation (9)
      $\psi_{t+1} \leftarrow \psi_t - \lambda_\pi \nabla_\psi \mathcal{J}(\pi, i)$
    **end for**
    $t \leftarrow t + T$
  **end for**
**end while**

**Algorithm 2** MOIQ with Preference Annotation (MOIQ-PA)

Initialize networks $Q_\phi$, $\pi_\psi$ and $\hat{P}_\theta$
**Input:** $\mathcal{D}_E$, a limited number of $\mathcal{D}_{LE}$
**while** environment step $t \leq$ N **do**
  **for** each $\omega_i$ **do**
    **for** each episode step in [1, T] **do**
      $a_t \sim \pi(\cdot|s_t, \omega_i)$
      $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, s_{t+1}, \omega_i)\}$
      Update $Q_\phi$ according to Equation (8)
      $\phi_{t+1} \leftarrow \phi_t + \lambda_Q \nabla_\phi \mathcal{J}(Q, i)$
      Update $\pi_\psi$ according to Equation (9)
      $\psi_{t+1} \leftarrow \psi_t - \lambda_\pi \nabla_\psi \mathcal{J}(\pi)$
    **end for**
    Update $\hat{P}_\theta$ according to Equation (10)
    $\theta_{t+1} \leftarrow \theta_t + \lambda_P \nabla_\theta \mathcal{J}(P, \tau)$
    $t \leftarrow t + T$
  **end for**
**end while**

**Critic network update:** We use $Q(s, a, \omega_i) \approx Q_i(s, a)$, which allows us to learn and estimate $Q$ value among various preferences. Starting with Equation (6), we fix $\pi$ and update the critic network:

$$\max_{Q} \mathcal{J}(Q,i) = \frac{1}{2}\cdot \mathbb{E}_{\rho_e}\Big[\phi\big(\hat{\boldsymbol{\omega}}^\top(\boldsymbol{Q}(s,a,\hat{\omega}) - \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\boldsymbol{V}^{\pi_e}(s',\hat{\omega}))\big)\Big]$$
$$+ \frac{1}{2}\cdot \mathbb{E}_{\rho_{\text{LE}}}\Big[\phi\big(\omega_{\text{LE}}^\top(\boldsymbol{Q}(s,a,\omega_{\text{LE}}) - \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\boldsymbol{V}^{\pi_E}(s',\omega_{\text{LE}}))\big)\Big] \tag{8}$$
$$- \mathbb{E}_{\tau\sim\mu,(s,a)\sim\tau}\big[\boldsymbol{\omega}^\top(\boldsymbol{V}^{\pi}(s) - \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\boldsymbol{V}^{\pi}(s'))\big] - \psi(\boldsymbol{\omega}_i^\top \boldsymbol{r}_i) - \beta\sum_{j=1}^{n}\|\boldsymbol{r}_i - \boldsymbol{r}_j\|_2$$

where $\boldsymbol{r_i} = \mathcal{T}^\pi Q_i$ is the estimated vector reward of $i$th agent. We choose $\mu$ to be the mixture distribution $\frac{1}{2}\rho_\pi + \frac{1}{4}\rho_E + \frac{1}{4}\rho_{LE}$. $\hat{\omega}$ is the predicted preference from posterior network conditioned on $\tau_E$, $\omega_{LE}$ is the labeled preference, and $\omega$ is the preference according to the chosen $\tau$.

**Actor network update:** We use $\pi(s,a,\omega_i) \approx \pi_i(s,a)$. For a fixed $Q$, we update $\pi$ by minimizing the expected KL-divergence (Haarnoja et al., 2018):

$$\min_{\pi} \mathcal{J}(\pi) = \mathbb{E}_{\tau\sim\mu,(s,a)\sim\tau}\big[\log \pi(a|s,\boldsymbol{\omega}) - \boldsymbol{\omega}^\top \boldsymbol{Q}(s,a,\boldsymbol{\omega})\big] \tag{9}$$

**Posterior network update:** By maximizing Equation (7) with respect to $\hat{P}$, we have:

$$\max_{\hat{P}} \mathcal{J}(\hat{P}) = \mathbb{E}_{\omega\sim p(\omega),\tau\sim\rho_\pi^\omega}[\log \hat{P}(\omega|\tau)] + \mathbb{E}_{\omega\sim p(\omega),\tau\sim\rho_E^\omega}[\log \hat{P}(\omega|\tau)] \tag{10}$$

### A.2 Environment Details

**DST:** DST environment is a classic MORL problem where the agent, controlling a submarine in a 2D world, observes a 2D continuous box with values in the range [0, 11] for both x and y coordinates. 2-dimensional reward space in the form *(treasure value, step cost)*, where treasure value follows (Yang et al., 2019) and step cost is $-1$ for each step.

**Mo-HalfCheetah:** Mo-HalfCheetah is a 2-dimensional robot with 9 body parts and 8 connecting joints. The goal is to apply torque on the joints to make the cheetah run forward (right) as fast as possible and minimize the control cost associated with each step taken. The system operates within a 17-dimensional observation space and a 6-dimensional action space. 2-dimensional reward space in the form *(velocity in x-axis, control cost)*.

**Mo-Walker:** Mo-Walker, characterized by a 6-degree-of-freedom bipedal robot with two legs and feet attached to a common base, operates within a 17-dimensional observation space and a 6-dimensional action space. The goal is to walk in the forward (right) direction by applying torques on six hinges connecting the seven body parts while minimizing the control cost. 2-dimensional reward space in the form *(velocity in x-axis, control cost)* with the healthy reward $+1$ is directly added to every dimension of reward if the agent is healthy at timestep $t$.

**Mo-Ant:** Mo-Ant, a 3D robot with a torso and four legs, operates in a 27-dimensional observation space and an 8-dimensional action space. It aims to coordinate four legs for forward motion while minimizing control costs. 2-dimensional reward space in the form *(velocity in x-axis, control cost)* with the healthy reward $+1$ is directly added to every reward dimension if the agent is healthy at timestep $t$.

**Mo-Hopper:** Mo-Hopper, a single-legged two-dimensional entity with four primary body segments, operates within an 11-dimensional observation space and a 3-dimensional action space. Its objective is to execute hops both in the forward (right) and in the upward direction by strategically applying torques to the three hinges connecting the body parts while mitigating control costs. 3-dimensional reward space in the form *(velocity in x-axis, height, control cost)* with the healthy reward $+1$ is directly added to every reward dimension if the agent is healthy at timestep $t$.

**Mo-Humanoid:**: Mo-Humanoid, a 3D bipedal robot designed for complex locomotion tasks, operates in a 378-dimensional observation space and a 17-dimensional action space. The agent must balance and coordinate its movements for efficient locomotion while minimizing energy costs. The reward space is 5-dimensional in the form (x-velocity, y-velocity, left elbow angle, right elbow angle, control cost). A healthy reward $+1$ is added to each dimension of the reward when the agent is healthy at timestep $t$.

### A.3 TRAINING CURVE AND IQM EVALUATION

**Training curve of MOIQ-PA on low-annotation regime:** As shown in Figure 4, MOIQ-PA significantly outperforms ESS-InfoGAIL in all environments except Mo-Hopper, highlighting its superior performance in most cases.

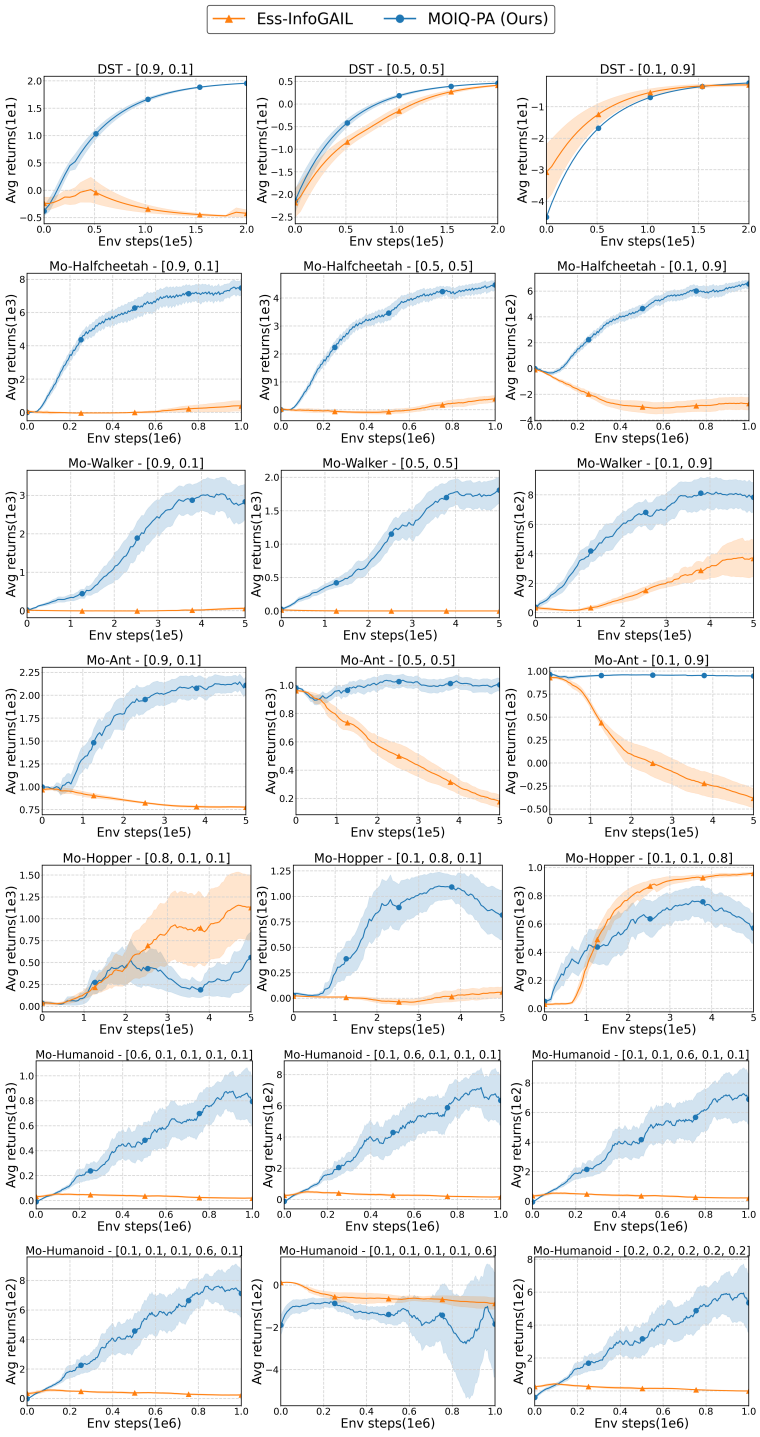



Figure 4: **Evaluation results while training.** The results are averaged from 5 seeds. All are smoothed by taking ewma return with *alpha*=0.1.

**IQM metric and performance profiles on low and annotation regime:** In our evaluation, we incorporate the Interquartile Mean (IQM) metric and performance profiles, both introduced by Agarwal et al. (2021). The IQM metric enhances robustness by computing the mean score over the middle 50% of run results, mitigating the impact of outliers. Meanwhile, performance profiles provide a comprehensive visualization of algorithmic performance by illustrating the distribution of normalized scores across environments.

As shown in Figure 5 and 6, our method consistently outperforms all baselines across both low-annotation and full-annotation settings, achieving normalized scores of 0.8 and 0.9, respectively. This demonstrates the effectiveness in both limited and fully supervised scenarios. Furthermore, the performance profiles highlight the stability of our approach. The smoother curve, in comparison to the baselines, indicates lower performance variance and greater reliability across tasks. This suggests that our method not only achieves higher average scores but also maintains robust and consistent performance across different runs.

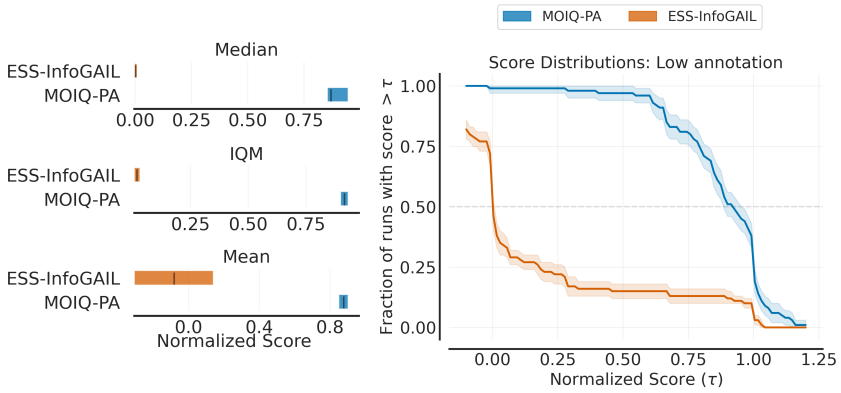



Figure 5: **IQM metric and performance profiles in the low-annotation regime. Left.** Showing the median, IQM, and mean scores under low-annotation. **Right.** Performance profiles under low-annotation based on score distributions

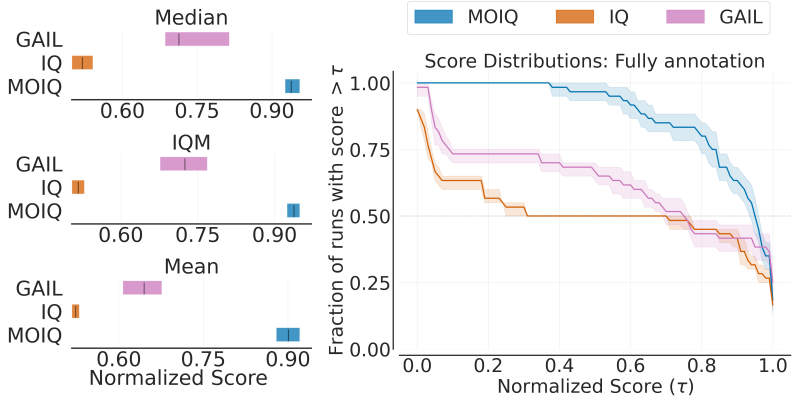



Figure 6: **IQM metric and performance profiles in the full-annotation regime. Left.** Showing the median, IQM, and mean scores under full-annotation. **Right.** Performance profiles under full-annotation based on score distributions.

### A.4 Additional Results

**Can MOIQ-PA handle demonstrations with noise?** We investigate the effect of weaker demonstrations by adding noise to the expert policies (this is achieved by randomly sampling an action with a 0.1 probability and otherwise following the original expert policy). As shown in Figure 7, despite the presence of noisy demonstrations, MOIQ-PA achieves performance that matches or exceeds that of experts, highlighting its robustness to demonstration quality.

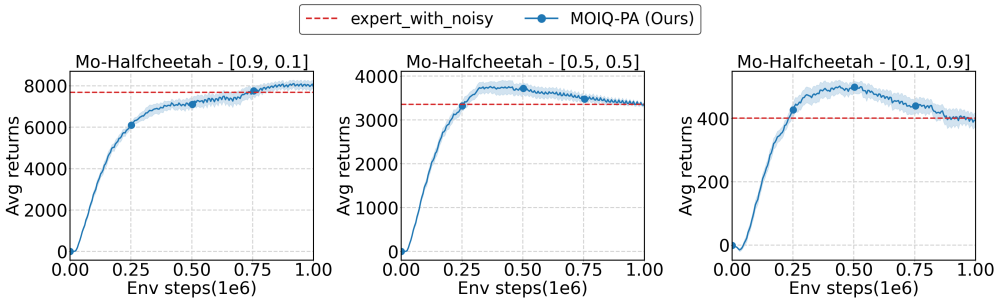



Figure 7: **Average return with weaker experts.** MOIQ-PA is able to approach and even exceed the performance of noisy experts. Results are averaged from 3 different seeds and smoothed by taking ewma return with *alpha*=0.1

**Can MOIQ-PA still perform well under an even lower-annotation regime?** To further demonstrate that the proposed MOIQ only requires a relatively small number of preference labels, we consider an even more challenging experimental scenario: For each preference, there are 100 expert trajectories but with only 1 of them is labeled (i.e., 1% annotation). As shown in Figure 8, the results show that MOIQ can still well imitate diverse expert behavior under only 1% annotation.

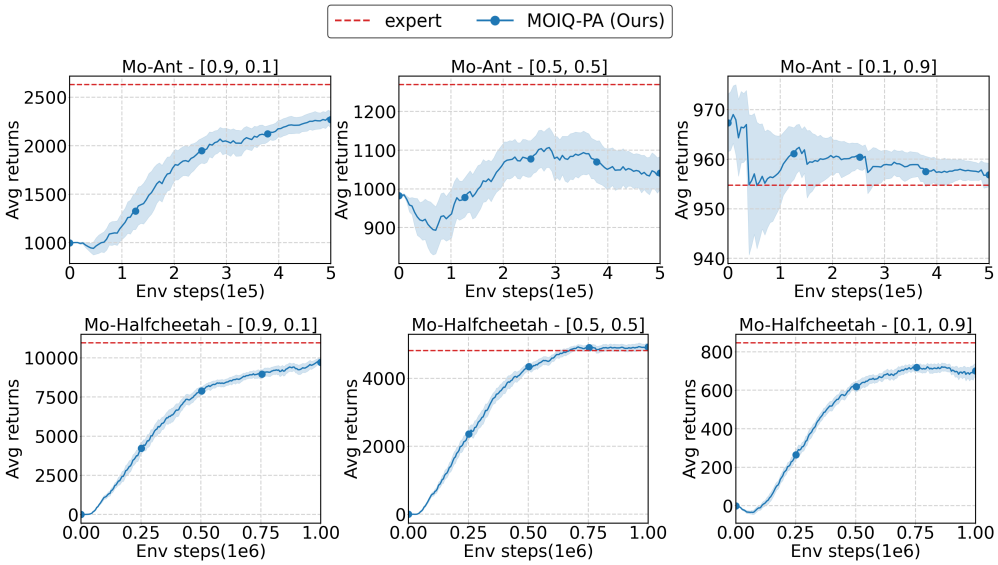



Figure 8: **Average return with 1%-annotation regime.** MOIQ-PA performs well in most cases, demonstrating its robustness. Results are averaged from 3 different seeds and smoothed by taking ewma return with *alpha*=0.1

**Can MOIQ-PA still well perform under the imbalanced amount of demonstrations?** We investigate six imbalanced scenarios where the number of trajectories for the preferences $[0.9, 0.1]$, $[0.5, 0.5]$, and $[0.1, 0.9]$ are denoted by $(x, y, z)$, respectively. We include six scenarios: $(1, 10, 10)$, $(10, 1, 10)$, $(10, 10, 1)$, $(10, 1, 1)$, $(1, 10, 1)$, and $(1, 1, 10)$, each preference contains only one trajectory labeled with ground-truth preference. We compare these imbalanced settings to the original balanced setting of $(10, 10, 10)$. As shown in Figure 9, we notice that in most imbalanced settings, given a specific preference, the results with 10 trajectories of that preference are generally better than the results with only 1 trajectory of that preference. The performance of MOIQ-PA remains remarkably stable across most imbalanced settings, highlighting its robustness to demonstration quantity, especially in imbalanced situations.

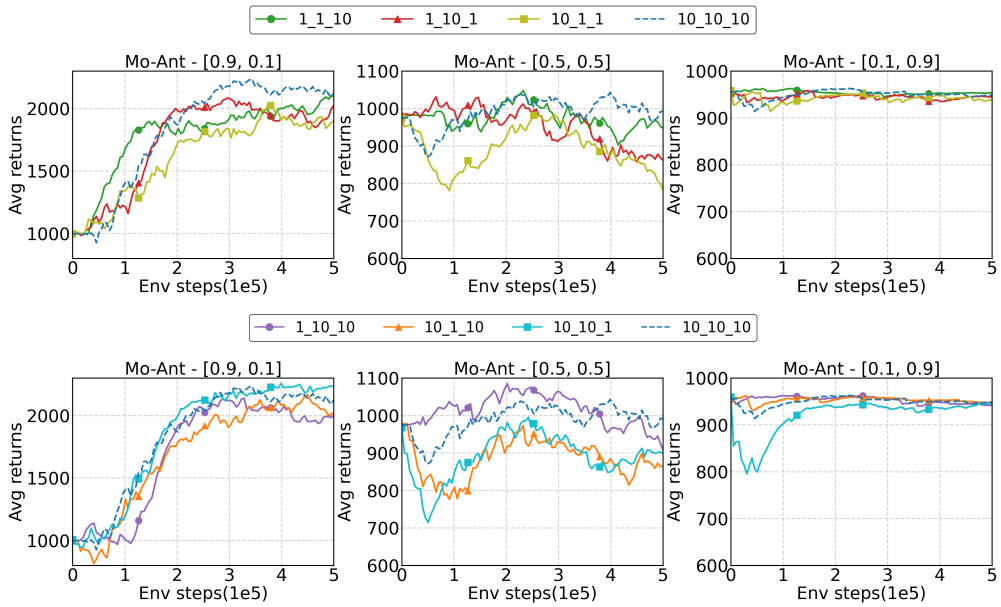



Figure 9: **Average return under imbalanced settings.** Each imbalanced setting is conducted over 1 seed and smoothed by taking ewma return with *alpha* $= 0.1$. While the evaluation preferences are identical across columns, they are intentionally separated for better readability.

**Does reward consensus actually bring benefits in transferbility?** We aim to quantify and analyze the benefits brought by reward consensus. Therefore, in addition to training the model with $\beta = 5$, we also trained the model with $\beta = 0$, representing the case without the reward consensus constraint. We present our ablation studies of the constraint coefficient by testing the transferability of these two models in Table 6.

The results indicate that the model with $\beta = 5$ exhibits significantly superior transferability compared to the model with $\beta = 0$, with the only exception being a minor disadvantage in EUM for Mo-Ant. Moreover, in the Mo-Hopper environment, $\beta = 5$ shows a clear advantage. To further illustrate this, we visualize the return of Mo-Hopper. Given that Mo-Hopper is in a 3-dimensional reward space, we assess transferability by fixing one dimension while varying the other two, resulting in three 2D projections, as shown in Figure 10. These visualizations demonstrate that $\beta = 5$ consistently provides a clear advantage, with well-defined transferability patterns observed in the Dimension 0 vs. Dimension 1 and Dimension 0 vs. Dimension 2 projections. In summary, our findings highlight that incorporating reward consensus significantly improves transferability across environments, particularly in complex reward spaces and unstable environments such as Mo-Hopper.

Table 6: **Evaluation of objective space coverage and reward optimization in the full-annotation regime.** The HV ($\times 10^3$) and EUM results for MOIQ-PA, InfoGAIL, and ESS-InfoGAIL across environments with a 2-dimensional reward space. The results are averaged across 5 different seeds, with boldface indicating the best performance. Additionally, for the Mo-Hopper environment, which has a 3-dimensional reward space, the HV is reported as $\times 10^7$.

| | Beta=5 | | Beta=0 | |
|---|---|---|---|---|
| Env | HV | EUM | HV | EUM |
| Mo-HalfCheetah | **3462.93** $\pm$ **156.72** | **5473.24** $\pm$ **155.09** | 3420.29 $\pm$ 104.07 | 5417.02 $\pm$ 108.62 |
| Mo-Walker | **4160.45** $\pm$ **200.11** | **2597.17** $\pm$ **88.64** | 3927.60 $\pm$ 76.80 | 2506.19 $\pm$ 25.16 |
| Mo-Ant | **9904.36** $\pm$ **151.73** | 1449.21 $\pm$ 26.12 | 9736.95 $\pm$ 464.99 | **1452.92** $\pm$ **43.21** |
| Mo-Hopper | **1772.06** $\pm$ **268.31** | **1841.25** $\pm$ **112.43** | 1129.97 $\pm$ 224.27 | 1536.20 $\pm$ 117.68 |

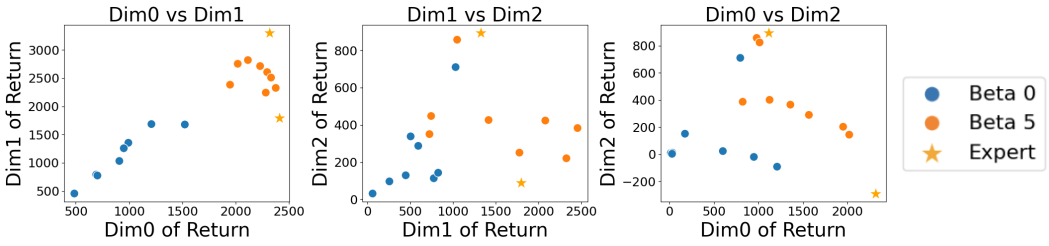



Figure 10: **Transferability of the best model on Mo-Hopper environment.** Each point is obtained by feeding in a specific preference value from $[0.8 - 0.1 \times i, 0.1 + 0.1 \times i, 0.1]$, $[0.1, 0.8 - 0.1 \times i, 0.1 + 0.1 \times i]$, $[0.8 - 0.1 \times i, 0.1, 0.1 + 0.1 \times i]$ for $i \in [1, 7]$. Evaluations are conducted over 100 episodes, and the results are averaged across 5 different seeds. The results for these three cases are presented in the following three figures.

### A.5 Ablation Study

**Ablation on window size:** This refers to the input length of the trajectory for our posterior network. Intuitively, a larger window size would lead to more accurate predictions, while a smaller window size is more flexible and can analyze smaller segments of a trajectory to determine which preference it aligns with. We evaluate MOIQ-PA under a window size of 10 and under the full trajectories.

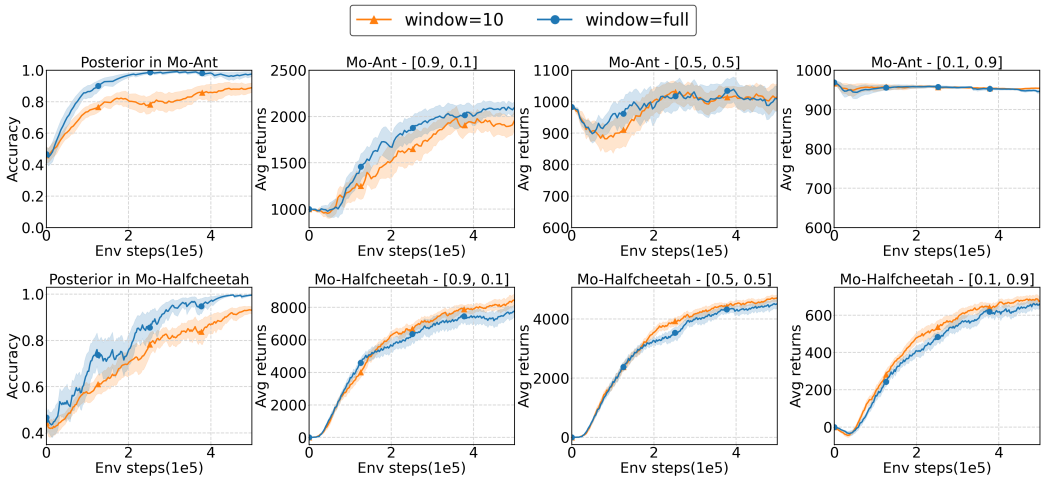



Figure 11: **Posterior accuracy and average return during training.** The full window size shows better posterior accuracy, while the performance shows mixed results. Results are averaged from 3 different seeds and smoothed by taking ewma return with *alpha*=0.1

### A.6 Additional Related Work

**Inverse Constrained Reinforcement Learning:** Early ICRL approaches focused on learning constraints from single-expert demonstrations using maximum entropy frameworks (Baert et al., 2023) and Bayesian methods (Papadimitriou et al., 2022), resulting robust policies but struggled with various conditions or types of multiple experts. (Qiao et al., 2024) addressed learning constraints from a mixture of expert demonstrations using flow-based density estimators for unsupervised agent identification. By incorporating contrastive learning, MMICRL captures diverse agent behaviors, improving constraint recovery and control performance with contrastive learning.

**Multi-task inverse RL:** Early work (Dimitrakakis & Rothkopf, 2012) extended IRL to multi-task scenarios from a Bayesian perspective, where each demonstration represented a different task, but struggled to scale to high-dimensional continuous state spaces. Maximum causal entropy-based approaches (Gleave & Habryka, 2018) improved on this by incorporating a regularization term in

Table 7: Comparison of different related work and their characteristics. The symbols used have the following meanings: ✓ indicates that the work fully supports the feature; ✗ indicates that the work does not support the feature; △ indicates that the featurework is partially supported.

| Method | Non-Adversarial Training | Heterogeneous | Meaningful latent factors | Recovers reward |
|---|---|---|---|---|
| GAIL-like methods/WAIL (Ho & Ermon, 2016) | ✗ | ✗ | ✗ | △ |
| Behavioral Cloning (Sasaki & Yamashina, 2020) | ✓ | ✗ | ✗ | ✗ |
| IQ-Learn (Garg et al., 2021) | ✓ | ✗ | ✗ | ✓ |
| ILEED (Beliaev et al., 2022) | ✓ | ✓ | △ | ✗ |
| InfoGAIL-like methodsrelated (Fu et al., 2023) | ✗ | ✓ | ✗ | △ |
| MOIQ (Ours) | ✓ | ✓ | ✓ | ✓ |

the loss function to reduce the number of demonstrations needed for reward recovery, and scaling to MDPs with infinite state spaces. Subsequently, Multi-task Hierarchical Adversarial IRL (Chen et al., 2023a) was introduced to learn hierarchical structures for multi-task policies, improving performance on complex tasks and demonstrating transferability without task annotations. Although multi-task IRL and multi-objective IRL both aim to increase the sample efficiency of IRL, multi-task IRL focuses on tasks across different environments, whereas multi-objective IRL involves heterogeneous demonstrations reflecting individual preferences in a consistent environment.

We summarize the characteristics of our method compared to related works in the table. Some works partially meet these characteristics. For example, GAIL (Ho & Ermon, 2016) and InfoGAIL (Li et al., 2017) do not directly recover the expert's reward function; instead, they learn a policy that approximates the expert's strategy by imitating the expert's behavior. ILEED's (Beliaev et al., 2022) latent factors are used to assess the expertise of trajectories, which differs from our definition of latent factors, and its assessment heavily relies on high-quality expert data.

### A.7 Compute resources and implementation details

**Compute resources:** For all our experiments, we run on both RTX 6000 Ada generation and RTX 4000 SFF Ada Generation GPU. For MOIQ, it took around 3 hours to train. For MOIQ-PA, it took around 24 and 8 hours for window size = 10 and full window size, respectively.

**Demonstrations:** For discrete DST, an optimal stochastic policy is adopted to collect demonstrations. Specifically, let $d_x^b$, $d_y^b$ be the distances to the border of the current grid along x and y axis, $d_x^\top$, $d_y^\top$ be the distances to the target treasure of the current grid along x and y axis. The probability of going right or down is proportional to the $\min(d_x^b, d_x^\top)$ and $\min(d_y^b, d_y^\top)$ of the current grid. For the continuous DST and Mujoco tasks (except for Mo-HalfCheetah), the experts are trained from scratch with SAC for each preference for 0.5M steps. For Mo-HalfCheetah, the experts are trained for 1.0M steps.

**Regularizers:** In most of our environments, we utilized the regularization term computed as the $\ell_2$-norm, which measures the deviation between the predicted and expected Q-values. This approach is inspired by the method in IQ-Learn. By leveraging this regularization, we ensured stable and precise updates to the policy.

In contrast, for the DST and HalfCheetah environments, we adopted a weighted regularization approach, expressed as $\psi(\omega_i^\top \cdot \boldsymbol{r}_i)$. This method introduces preference-based adjustments, enhancing the distinction between different preferences within the system. By incorporating these preferences, the model is able to better align its learning objectives with the nuances of the environment.

**GAIL:** We implement GAIL using imitation (Gleave et al., 2022), with SAC as the generator.

**MOIQ and Experts in continuous DST and Mujoco tasks (except for Mo-HalfCheetah):** We implement our algorithm based on the open-source code of IQ-Learn (Garg et al., 2021). Its implementation is built on SAC, the hyperparameters used are listed in table 9.

**Experts in Mo-HalfCheetah and Mo-Humanoid:** To train stronger experts on Mo-HalfCheetah and Mo-Humanoid, we specifically implemented SAC from the open-source code of CleanRL (Huang et al., 2022). The hyperparameters used are listed in table 10.

Table 8: **Hyperparameters of GAIL**

| Parameter | Value |
|---|---|
| Policy | MlpPolicy |
| Learning rate | 3e-4 |
| Buffer size | 1e6 |
| Batch size | 256 |
| Tau | 0.05 |
| Gamma | 0.99 |
| Train frequency | 1 |

Table 9: **Hyperparameters of MOIQ**

| Parameter | Value |
|---|---|
| Policy | MlpPolicy |
| Hidden dim | [255, 255] |
| Critic lr | 3e-4 |
| Actor lr | 3e-5 |
| Buffer size | 1e6 |
| Batch size | 256 |
| Critic update frequency | 1 |
| Actor update frequency | 1 |
| Critic tau | 0.005 |

**InfoGAIL and Ess-InfoGAIL:** We implement our algorithm based on the open-source code of Ess-InfoGAIL (Fu et al., 2023). In Ess-InfoGAIL, the hyperparameters used are listed in table 11. In InfoGAIL, we remove the latent skill variable from Ess-InfoGAIL and eliminate the semi-supervised encoder update. Instead, we set the latent code to be a one-hot encoded vector with 3 dimensions. The hyperparameters used are listed in table 12.

Table 10: **Hyperparameters of experts in Mo-HalfCheetah and Mo-Humanoid**

| Parameter | Value |
|---|---|
| Policy | MlpPolicy |
| Hidden dim | [255, 255] |
| Critic lr | 1e-3 |
| Actor lr | 3e-4 |
| Buffer size | 1e6 |
| Batch size | 256 |
| Critic update frequency | 1 |
| Actor update frequency | 2 |
| Critic tau | 0.005 |

Table 11: **Hyperparameters of Ess-InfoGAIL**

| Parameter | Value |
|---|---|
| Optimizer | Adam |
| Hidden dim | [100, 100] |
| Policy/Value learning rate | 0.003 |
| Discriminator learning rate | 0.005 |
| Encoder learning rate | 0.01 |
| Batch size | 1000 |
| Policy/Value update iterations | 20 |
| Discriminator/Encoder iterations | 50 |
| Gumbel-Softmax temperature tau | 0.1 |
| Weighting coefficient lamda1 | 1.0 |
| Weighting coefficient lamda2 | 4.0 |
| Weighting coefficient lamda3 | 3.0 |

Table 12: **Hyperparameters of InfoGAIL**

| Parameter | Value |
|---|---|
| Optimizer | Adam |
| Hidden dim | [100, 100] |
| Policy/Value learning rate | 0.003 |
| Discriminator learning rate | 0.005 |
| Encoder learning rate | 0.01 |
| Batch size | 1000 |
| Policy/Value update iterations | 20 |
| Discriminator/Encoder iterations | 50 |
| Weighting coefficient lamda1 | 1.0 |