# Learning Fairness from Demonstrations via Inverse Reinforcement Learning

Jack Blandin
blandin1@uic.edu
University of Illinois Chicago,
Department of Computer Science
Chicago, IL, USA

Ian Kash
iankash@uic.edu
University of Illinois Chicago,
Department of Computer Science
Chicago, IL, USA

## ABSTRACT

Defining fairness in algorithmic contexts is challenging, particularly when adapting to new domains. Our research introduces a novel method for learning and applying group fairness preferences across different classification domains, without the need for manual fine-tuning. Utilizing concepts from inverse reinforcement learning (IRL), our approach enables the extraction and application of fairness preferences from human experts or established algorithms. We propose the first technique for using IRL to recover and adapt group fairness preferences to new domains, offering a low-touch solution for implementing fair classifiers in settings where expert-established fairness tradeoffs are not yet defined.

## CCS CONCEPTS

• **Computing methodologies** → **Inverse reinforcement learning**; *Supervised learning by classification*; • **Social and professional topics** → **Computing / technology policy**.

## KEYWORDS

inverse reinforcement learning, group fairness in classification, fairness transfer learning

**ACM Reference Format:**
Jack Blandin and Ian Kash. 2024. Learning Fairness from Demonstrations via Inverse Reinforcement Learning. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24), June 03–06, 2024, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3630106.3658539

## 1 INTRODUCTION

Achieving algorithmic fairness in new domains presents a distinct challenge. Unlike standard performance metrics like precision and recall, where it is relatively simple to select the appropriate measure, selecting the right fairness metric is far more nuanced. It involves striking a delicate balance between multiple, often conflicting [17], desiderata in order to find the appropriate level of trade-offs for the given domain. Current practices largely rely on manual fine-tuning to adapt fairness criteria for each new context, which is a significant burden to implementing fairness in practice [19]. This work introduces a novel framework for extracting and applying domain-agnostic fairness objectives, or *fairness preferences*, learned from either human or algorithmic experts, to data exhibiting covariate shifts or residing in entirely new domains. Our approach reduces the need for manual fine-tuning, fostering transferable fairness criteria.

We leverage concepts from inverse reinforcement learning (IRL) to infer an expert's implicit fairness preferences from their past actions or "demonstrations." Consider, for instance, a university admissions board recognized for its fair practices. By analyzing their historical decisions, we can extract their inherent fairness principles guiding their choices, and then use these principles to design fair policies in new contexts. This mirrors the IRL objective: given a policy (in this case, the admissions decisions), uncover the underlying reward function (fairness preferences) that informs it. Our central inquiry is whether IRL techniques can effectively capture expert group fairness preferences through such demonstrations, and whether these learned preferences can subsequently be used to train classifiers that exhibit fairness in new domains. To the best of our knowledge, this paper represents the first attempt to do so.

We consider the following scenario: given a classification dataset, and a set expert decisions (i.e. predictions), try to recover the expert's fairness preferences so that we can use them to compute new classifiers that generalize these fairness preferences to new datasets. Following the *feature matching* technique of Abbeel and Ng's classical IRL approach [1], we seek a reward function (fairness preferences) who's optimal classifier behaves "similarly" to the expert, where "similarly" refers to how similar they perform on a set of fairness measures and related metrics.

Our objective is not to provide the "best" IRL approach to learning group fairness preferences. Rather, the focus of this work is to answer the following question: is it feasible to use IRL to extract and transfer fairness knowledge, embodied in algorithmic or human demonstrations, to train fair classifiers in domains where fairness has not yet been explicitly defined? To this end, we offer two primary technical contributions. First, in Section 3, we propose a novel method that frames group fairness in classification as an IRL problem, with the reward function's weights reflecting experts' fairness preferences. Second, in Section 4, we present an algorithm inspired by the classical max-margin IRL approach of Abbeel and Ng [1]. Our algorithm effectively extracts fairness preferences from expert demonstrations and uses them to train a new classifier on either the original demonstrator domain or entirely new domains. In


*FAccT '24, June 03–06, 2024, Rio de Janeiro, Brazil*

ACM ISBN 979-8-4007-0450-5/24/06
https://doi.org/10.1145/3630106.3658539

Section 5 we show through experiments on the Adult, ACSIncome, and Boston Housing datasets that our approach can (i) recover expert fairness preferences, (ii) leverage these preferences to learn fair policies robust to covariate shift, and (iii) transfer these learned fairness preferences to learn fair policies in entirely new domains. In Section 6 we discuss the limitations of our work, outline promising avenues for future research, and provide concluding remarks.

### 1.1 Related Work

*Preference Elicitation.* Our objective relates to that of preference elicitation [6], in which a user is repeatedly asked to decide on a pair of choices to help the user efficiently define their preferences. Preference elicitation has been studied in the fairness setting [14, 15] to learn fairness objectives from noisy user feedback. Our work differs in that we do not assume that there is a user to repeatedly solicit feedback from in the domain which we aim to implement fairness.

*Fairness tradeoffs.* There is often debate around which fairness metric is relevant to a particular application. For instance, Kleinberg et al. [17] show that it is not possible to simultaneously satisfy several common group fairness metrics. Therefore, finding efficient tradeoffs between fairness metrics is a desirable quality in any fairness algorithm. Our approach works well in this environment by identifying the fairness tradeoffs made by an existing algorithm, which could be used to extend the algorithm to more settings, or purely as insight into the objectives of the decision-maker.

*IRL and Imitation Learning.* Our work leverages IRL [1, 3, 18, 26], which recovers a reward function based on a set of policy demonstrations in an MDP environment. Our work also relates to *imitation learning*, where the goal is to learn a policy that imitates an expert by observing their demonstrations. Memarrast et al. [21] provide an imitation learning approach that leverages subdominance minimization [25] to improve on the decisions of a suboptimal human trying to achieve fairness. However, their technique does not attempt to recover the fairness objectives of the demonstrator, which prevents it from being used to transfer fairness preferences.

*Learning fair representations.* Madras et al. [20] use adversarial methods to learn fair data representations that are robust to third-party vendors with unknown objectives learning predictors from the data. Oneto et al. [22] expand on this by using multitask learning to enhance how fairness generalizes over varied applications. Both approaches share our focus on generalizing fairness to new contexts.

*Covariate shift.* When training a classifier on a set of training data, it is typically assumed that the training distribution matches the test distribution. Covariate shift [24] is the problem when this assumption does not hold, and the distribution changes between the training and test data. Prior work has addressed this by training a fair classifier using a robust optimization approach [23] or weight perturbation [16]. IRL provides an alternative by learning a portable representation of the classifier's objectives which allows for training an alternative in the face of covariate shifts, even with only black-box access to the original classifier.

## 2 PRELIMINARIES

Here we provide prerequisite information for the two domains we marry: group fairness in classification and IRL.

### 2.1 Group Fairness in Classification

We focus on binary classification tasks. Following Hardt et al. [12], we consider an $n$-record dataset drawn iid from a population represented by the joint distribution of $(Z, X, Y)$ where each record $(z_i, x_i, y_i)$ represents a single individual and consists of a binary sensitive attribute $z_i \in \{0, 1\}$; a $k$-length vector of discrete, non-sensitive attributes $x_i \in \mathbb{Z}^k$; and a binary label $y_i \in \{0, 1\}$. The classification goal is to fit a classifier $C(Z, X) = \hat{Y} \in \{0, 1\}$ that maximizes some measure of efficiency. For instance, a common efficiency metric is to maximize accuracy, which we can model as a *loss function* $L : (Z, X, Y, \hat{Y}) \to \mathbb{R}$:

$$L_{\mathsf{Acc}}(Z, X, Y, \hat{Y}) = -\frac{1}{n} \sum_{i=0}^{n-1} \mathbb{1}_{\hat{y}_i = y_i} \ . \tag{1}$$

The goal in *group fairness* is to also optimize for one or more fairness metrics in addition to efficiency. For instance, one common group fairness metric is *demographic parity*, which requires the probability that an individual receives a positive prediction be independent of their protected attribute:

$$P(\hat{Y} = 1 \mid Z = 0) = P(\hat{Y} = 1 \mid Z = 1) \ . \tag{2}$$

Similar to demographic parity is *equal opportunity* [12], which adds a qualification condition to the constraint:

$$P(\hat{Y} = 1 \mid Y = 1, Z = 0) = P(\hat{Y} = 1 \mid Y = 1, Z = 1) \ . \tag{3}$$

### 2.2 MDPs

In order to understand how IRL algorithms work, which our own techniques build on, it is helpful to understand the domain they are defined for: Markov Decision Processes (MDPs). An MDP consists of a six-tuple $(S, A, T, \gamma, \rho, R)$, where $S$ is a set of states, $A$ is a set of actions, $T(s, a, s')$ gives the probability of landing in state $s'$ after taking action $a$ in state $s$, $\gamma \in [0, 1]$ is a discount rate for future utility, $\rho$ is a probability distribution over the initial state where $\rho(s)$ is the probability of starting in state $s$, and $R(s, a)$ is the reward for taking action $a$ in state $s$. The goal of an MDP is to find a policy $\pi : S \to A$ that maximizes the expected discounted sum of rewards. There are many techniques for finding or approximating optimal policies, but the most relevant one for our purposes is linear programming. The following simple linear program allows us to find an optimal policy given rewards $R$:

$$\begin{aligned} \lambda^* \leftarrow \underset{\lambda}{\operatorname{argmax}} & \sum_{s \in S} \sum_{a \in A} \lambda(s, a) R(s, a) \\ \text{s.t. } & \lambda(s', a') \geq 0 \ \forall s' \in S, a' \in A; \\ \sum_{a' \in A} \lambda(s', a') &= \rho(s') + \gamma \sum_{s \in S} \sum_{a \in A} \lambda(s, a) T(s, a, s') \ \forall s' \in S \end{aligned} \tag{4}$$

where $\lambda(s, a)$ is the occupancy measure for state $s$ and action $a$. We compute the optimal policy directly from $\lambda^*$.

### 2.3 IRL

IRL infers a reward function directly from an MDP policy. Given an expert demonstration policy $\pi^E$, [1] the objective is to construct

[1] Demonstrations may also be in the form of trajectories.

the expert's unknown reward function $R^E$ that $\pi^E$ is optimizing. Many IRL algorithms assume that

$$R^E(s) = w^E \cdot \phi(s) \tag{5}$$

where $\phi : S \to \mathbb{R}^k$ maps each state to a $k$-length vector of features and $w^E$ is a vector of feature weights. The features $\phi$ are curated manually or algorithmically to reduce the dimensionality of the state space, which makes IRL more tractable. Their more relevant representation is their expected discounted accumulated value vector under some policy $\pi$. These are referred to more succinctly as the policy's *feature expectations* $\mu(\pi)$, which are defined as

$$\mu(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) \mid \pi] \in \mathbb{R}^k \ . \tag{6}$$

Therefore, a policy's feature expectations refer to how often the policy encounters each of the $\phi$ features. So in some sense they are a compact description of the policy, which make them useful for IRL approaches that use *feature matching* to recover the expert's reward. Feature matching works as follows: in order to recover the expert's unknown reward function $R^E$, find a candidate reward function $R^L$ who's optimal policy $\pi^L$ behaves similarly to the expert's demonstrations $\pi^E$, where "similarly" is defined in terms of the difference in feature expectations. Finding such a reward then reduces to finding weights $w^L$ such that their optimal policy achieves feature expectations $\mu(\pi^L)$ similar to those of the expert $\mu(\pi^E)$:

$$w^L \leftarrow \operatorname*{argmin}_{w} \operatorname*{argmin}_{\pi} w \cdot [\mu(\pi^E) - \mu(\pi)] \ . \tag{7}$$

Abbeel and Ng's approach [1], which we build on, implements Equation 7 with a loop where each iteration learns a new policy by maximizing a candidate $w^L$. In our setting, this can be computed using Equation 4 and an LP-solver.

## 3 MODELING GROUP FAIRNESS AS IRL

Here we provide our first primary contribution: a way to model group fairness in classification as an IRL problem where the learned reward weights represent the expert's fairness and efficiency preferences.

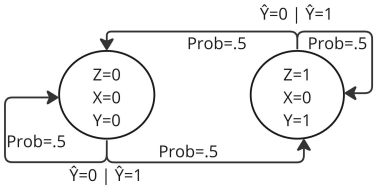
### 3.1 Problem Statement

In this paper we assume that the desired notion of fairness can be demonstrated by a human or algorithmic expert, or that they already exist as a dataset. We assume the expert is optimizing for at least one efficiency metric, at least one fairness metric, or a combination of efficiency and fairness metrics. The expert provides demonstrations by making $m$ predictions $\hat{Y}$ on a set of inputs $(Z, X)$ sampled from an $n$-record classification dataset $(Z, X, Y)$. Our objective is to recover the expert's preferences from a set of *demonstrations* $(Z, X, Y, \hat{Y})$.

### 3.2 Group Fairness as Weighted Rewards

We desire to use IRL techniques to recover the expert's fairness and efficiency preferences from their demonstrations. In order to do this, we represent the group fairness in classification problem as a single reward function $R(Z, X, Y, \hat{Y}) \to \mathbb{R}$. As a representative example, suppose our efficiency objective is to optimize accuracy, and our fairness objectives are to optimize demographic parity and equal opportunity. This corresponds to minimizing Equation (1) subject to Equations (2) and (3). We can represent this as a single reward function by representing demographic parity and equal opportunity constraints as minimizations of their violations, and combining them with Equation (1) as a single linear weighted sum:

**Figure 1: A simple two-record group fairness classification dataset represented as an MDP. The left circle represents one record $(z_0, x_0, y_0) = (0, 0, 0)$ and the right circle the other $(z_1, x_1, y_1) = (1, 0, 1)$. State transitions correspond to a new record being sampled.**




$$R(Z, X, Y, \hat{Y}) = w_1 R_{\mathsf{Acc}} + w_2 R_{\mathsf{DemPar}} + w_3 R_{\mathsf{EqOpp}} \tag{8}$$

$$\text{where } R_{\mathsf{Acc}}(Z, X, Y, \hat{Y}) = \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{1}_{\hat{y}_i = y_i} \ , \tag{9}$$

$$R_{\mathsf{DemPar}}(Z, X, Y, \hat{Y}) = -\left| \frac{\sum_{i=0}^{n-1} \mathbb{1}_{\hat{y}_i=1 \wedge z_i=0}}{\sum_{i=0}^{n-1} \mathbb{1}_{z_i=0}} - \frac{\sum_{i=0}^{n-1} \mathbb{1}_{\hat{y}_i=1 \wedge z_i=1}}{\sum_{i=0}^{n-1} \mathbb{1}_{z_i=1}} \right| , \tag{10}$$

$$R_{\mathsf{EqOpp}}(Z, X, Y, \hat{Y}) = -\left| \frac{\sum_{i=0}^{n-1} \mathbb{1}_{\hat{y}_i=1 \wedge y_i=1 \wedge z_i=0}}{\sum_{i=0}^{n-1} \mathbb{1}_{y_i=1 \wedge z_i=0}} - \frac{\sum_{i=0}^{n-1} \mathbb{1}_{\hat{y}_i=1 \wedge y_i=1 \wedge z_i=1}}{\sum_{i=0}^{n-1} \mathbb{1}_{y_i=1 \wedge z_i=1}} \right| . \tag{11}$$

Here, $w_1$ represents the linear weight for accuracy, $w_2$ for demographic parity, and $w_3$ for equal opportunity. Next we need to decide on the values for each weight. This setup resembles that of IRL—notice the similarity between Equation (5) and (8) . The features $\phi$ in Equation (5) parallel the efficiency and fairness metrics in Equation (8). Therefore, if we are able to model our efficiency and fairness metrics as features $\phi$, we can compute their feature expectations for the expert's demonstrations $\mu(\pi^E)$, which we can use to recover $w$ via Equation (7).

### 3.3 Fairness Feature Expectations

In order to find the expert's efficiency and fairness preferences, we need to define feature expectations such that policies with matching expectations exhibit similar fairness measures. These feature expectations can include both explicit fairness measures like demographic parity (positive rate parity) and complementary metrics; not technically fairness metrics but components of fairness measures, such as the positive rate for each protected group. Furthermore, in order to enable our IRL Algorithm (Section 4) that makes use of linear programming, we need the feature expectations to be linear functions of $\lambda$.

To produce these feature expectations, one might consider attempting to construct features $\phi$ whose expected discounted sums (Equation (6)) yield the desired metrics. For composite feature expectations, such as group positive rate ($P(\hat{Y} \mid Z = z) \forall\ z \in \{0, 1\}$), this is no issue. For most group fairness metrics, however, this is not possible since they contain an absolute value, which breaks the linearity assumption. Fortunately, most fairness metrics *can* be represented directly as feature expectations, rather than the expected discounted sums of features, which is ultimately the goal anyway.

**Algorithm 1** `FairIRL` learns the expert's fairness preferences from a set of demonstrations. From these learned preferences, `FairIRL` computes a new classifier for either the expert's domain, or an entirely new domain.

**Input:** $(Z, X, Y, \hat{Y}^E)$ ▷ Expert demonstrations on source dataset
**Input:** FeatExpGen ▷ Feature expectation generator
**Input:** $(Z', X', Y')$ ▷ Target dataset
**Output:** $w^{L^*}$ ▷ Learned reward weights
**Output:** $C^{L'}$ ▷ Learned classifier for target dataset

$\mu^E \leftarrow \text{FeatExpGen}(Z, X, Y, \hat{Y}^E)$ ▷ Compute expert feature expectations
$\mathbf{C^L}[0] \leftarrow$ a random initial classifier ▷ Generate any classifier as a starting point for learner
$\hat{Y}^L \leftarrow \mathbf{C^L}[0](Z, X)$ ▷ Generate predictions for initial learner classifier
$\mu^{\mathbf{L}}[0] = \text{FeatExpGen}(Z, X, Y, \hat{Y}^L)$ ▷ Generate feature expectations for initial learner classifier
$\mathbf{t}[0] \leftarrow \infty$ ▷ Initialize error array, and set initial value to infinity
$i \leftarrow 0$ ▷ Initialize counter
**while** $\mathbf{t}[i] > \epsilon \wedge \text{argmin}(t) = i$ **do** ▷ Stop if error below $\epsilon$ convergence threshold or if error starts increasing
  $\mathbf{X}_{\text{IRL}} \leftarrow \{\mu^E, \mu^{\mathbf{L}}[0], ..., \mu^{\mathbf{L}}[i]\}$ ▷ SVM inputs are expert and learned feat. exp.
  $\mathbf{Y}_{\text{IRL}} \leftarrow \{1, 0, ..., 0\}$ ▷ Expert feat. exp. are the "ones", learned are "zeros"
  $\texttt{SVM} \leftarrow \texttt{SVM.Fit}(X_{\text{IRL}}, y_{\text{IRL}})$ ▷ Fit the SVM to distinguish $\mu^E$ from $\mu^{\mathbf{L}}$
  $w^L[i] \leftarrow \texttt{SVM.Weights}$ ▷ Extract weights from SVM that best separate $\mu^E$ from $\mu^{\mathbf{L}}$
  $\mathbf{t}[i+1] \leftarrow \texttt{SVM.Margin}$ ▷ Set the SVM hyperplane margin as this iteration's error
  $\mathbf{C^L}[i+1] \leftarrow \text{LP}(\mathbf{w^L}[i], (Z, X, Y))$ ▷ Solve LP for $\mathbf{w^L}[i]$ on source data
  $\hat{Y}^L \leftarrow C^L[i+1](Z, X)$ ▷ Generate predictions from learned classifier
  $\mu^{\mathbf{L}}[i+1] \leftarrow \text{FeatExpGen}(Z, X, Y, \hat{Y}^L)$ ▷ Generate feature expectations for $\mathbf{C^L}[i+1]$
  $i \leftarrow i + 1$ ▷ Increment counter
**end while**
$i^* \leftarrow \text{argmin}(t)$ ▷ Find the iteration with the lowest error
$w^{L^*} \leftarrow w^L[i^*]$ ▷ The weights with the lowest error are the final weights $w^{L^*}$
$C^{L'} \leftarrow \text{LP}(w^{L^*}, \mathbf{K}, (Z', X', Y'))$ ▷ Solve LP for $w^{L^*}$ and feat. exp. constraints $\mathbf{K}$ on target dataset

In other words, there is no $\phi$ such that we can represent absolute value fairness metrics as $\mu = \mathbb{E}[\sum_{z,x,y,\hat{y}} \phi(z, x, y, \hat{y})]$, but we *can* represent them as expectations of some function over the entire set of demonstrations $\mu = \mathbb{E}[f(Z, X, Y, \hat{Y})]$. We can do this by representing them directly as state-action stationary distributions $\lambda$ of a special-case MDP. This special case MDP is a classification problem, specifically a group fairness classification problem, represented in MDP form. In short, this special-case MDP is characterized by single-step trajectories ($\gamma = 0$), where each state transition is a sample $(z_i, x_i, y_i)$ from the classification dataset $(Z, X, Y)$, and the MDP actions correspond to classifier predictions $C(z_i, x_i) \rightarrow \hat{y}_i$. This simplifies the second constraint from Equation (4), resulting in a new linear program:

$$\begin{aligned} \lambda^* \leftarrow \underset{\lambda}{\text{argmax}} & \sum_s \sum_a \lambda(s,a) R(s,a) \\ \text{s.t.} \quad & \lambda(s', a') \geq 0 \;\forall\; s' \in S, a' \in A; \\ & \sum_{a'} \lambda(s', a') = \rho(s') \;\forall\; s' \in S\ . \end{aligned} \tag{12}$$

Figure 1 shows a simple example of this special-case MDP.

Next, we demonstrate two examples of how to represent efficiency and fairness metrics as linear combinations of $\lambda$. We do so for one efficiency metric (accuracy) and one fairness metric (equal opportunity). Starting with accuracy, we can rewrite Equation (9) using the $\lambda(s, a)$ variables from Equation (12) as

$$\mu_{\texttt{Acc}} = \sum_{i=0}^{n-1} \lambda(s_i, a_i) \tag{13}$$

where $s_i = (z_i, x_i, y_i)$ and $a_i = y_i$. Thus $\mu_{\texttt{Acc}}$ is linear in $\lambda$ as desired. However, as previously mentioned, many group fairness notions are inherently non-linear. Using equal opportunity as a representative example, we can rewrite Equation (11) to get

$$\mu_{\texttt{Eq0pp}} = -\left| \frac{\sum\limits_{\{i|y_i=0 \wedge z_i=0\}} \lambda(s_i, 1)}{\sum\limits_{\{i|y_i=0 \wedge z_i=0\}} \sum\limits_a \lambda(s_i, a)} - \frac{\sum\limits_{\{i|y_i=0 \wedge z_i=1\}} \lambda(s_i, 1)}{\sum\limits_{\{i|y_i=0 \wedge z_i=1\}} \sum\limits_a \lambda(s_i, a)} \right| . \tag{14}$$

We can simplify this by observing that the denominators are independent of $\lambda$ since they are proportional to the number of qualified individuals from each group, which are constants. However, this still leaves a non-linearity due to the absolute value for a given dataset. Replacing the Equation (14) denominators with the appropriate constants, we can rewrite $\mu_{\texttt{Eq0pp}}$ as:

$$\begin{aligned} \min\Bigg( & c_0 \sum_{\{i|y_i=0 \wedge z_i=0\}} \lambda(s_i, 1) - c_1 \sum_{\{i|y_i=0 \wedge z_i=1\}} \lambda(s_i, 1), \\ & c_1 \sum_{\{i|y_i=0 \wedge z_i=1\}} \lambda(s_i, 1) - c_0 \sum_{\{i|y_i=0 \wedge z_i=0\}} \lambda(s_i, 1) \Bigg). \end{aligned} \tag{15}$$

While this "feature expectation" is non-linear, it is the minimum of two linear terms. Thus we can solve the mathematical program from Equation (12) by using a standard trick to encode a minimum in a linear program. So we add $\mu_{\texttt{Eq0pp}}$ as a variable to our Equation (12) linear program along with the following two constraints:

$$\begin{aligned} \mu_{\texttt{Eq0pp}} & \leq c_0 \sum_{\{i|y_i=0, z_i=0\}} \lambda(s_i, 1) - c_1 \sum_{\{i|y_i=0, z_i=1\}} \lambda(s_i, 1)\ , \\ \mu_{\texttt{Eq0pp}} & \leq c_1 \sum_{\{i|y_i=0, z_i=1\}} \lambda(s_i, 1) - c_0 \sum_{\{i|y_i=0, z_i=0\}} \lambda(s_i, 1)\ . \end{aligned} \tag{16}$$

Summarizing, we represent equal opportunity as the minimum of two linear feature expectations, which integrates easily with our

**Table 1: Feature expectations and performance measures used in our experiments.**

| Shorthand | Description | Definition | Feat Exp? |
|---|---|---|---|
| `Acc` | Accuracy | $P(\hat{Y} = Y)$ | Yes |
| `DemPar` | Demographic Parity (positive rate parity) | $1 - \lvert P(\hat{Y} = 1 \mid Z = 0) - P(\hat{Y} = 1 \mid Z = 1)\rvert$ | Yes |
| `EqOpp` | Equal Opportunity (true positive rate parity) | $1 - \lvert P(\hat{Y} = 1 \mid Z = 0, Y = 1) - P(\hat{Y} = 1 \mid Z = 1, Y = 1)\rvert$ | Yes |
| `TNRPar` | True Negative Rate Parity | $1 - \lvert P(\hat{Y} = 0 \mid Z = 0, Y = 0) - P(\hat{Y} = 0 \mid Z = 1, Y = 0)\rvert$ | Yes |
| `PR_Z0` | Positive Rate for the $Z = 0$ group | $P(\hat{Y} = 1 \mid Z = 0)$ | Yes |
| `PR_Z1` | Positive Rate for the $Z = 1$ group | $P(\hat{Y} = 1 \mid Z = 1)$ | Yes |
| `NR_Z0` | Negative Rate for the $Z = 0$ group | $P(\hat{Y} = 0 \mid Z = 0)$ | Yes |
| `NR_Z1` | Negative Rate for the $Z = 1$ group | $P(\hat{Y} = 0 \mid Z = 1)$ | Yes |
| `TPR_Z0` | True Positive Rate for the $Z = 0$ group | $P(\hat{Y} = 1 \mid Z = 0, Y = 1)$ | Yes |
| `TPR_Z1` | True Positive Rate for the $Z = 1$ group | $P(\hat{Y} = 1 \mid Z = 1, Y = 1)$ | Yes |
| `TNR_Z0` | True Negative Rate for the $Z = 0$ group | $P(\hat{Y} = 0 \mid Z = 0, Y = 0)$ | Yes |
| `TNR_Z1` | True Negative Rate for the $Z = 1$ group | $P(\hat{Y} = 0 \mid Z = 1, Y = 0)$ | Yes |
| `FPRPar` | False Positive Rate Parity | $1 - \lvert P(\hat{Y} = 1 \mid Z = 0, Y = 0) - P(\hat{Y} = 1 \mid Z = 1, Y = 0)\rvert$ | No |
| `FNRPar` | False Negative Rate Parity | $1 - \lvert P(\hat{Y} = 0 \mid Z = 0, Y = 1) - P(\hat{Y} = 0 \mid Z = 1, Y = 1)\rvert$ | No |
| `PrPar` | Predictive Parity | $1 - \lvert P(Y = 1 \mid Z = 0, \hat{Y} = 1) - P(Y = 1 \mid Z = 1, \hat{Y} = 1)\rvert$ | No |
| `NPrPar` | Negative Predictive Parity | $1 - \lvert P(Y = 0 \mid Z = 0, \hat{Y} = 0) - P(Y = 0 \mid Z = 1, \hat{Y} = 0)\rvert$ | No |

linear program in Equation (12). Although we provided an example for equal opportunity, this approach extends to many other group fairness notions, including demographic parity, equalized odds [12], false positive and false negative error rate balance [7], accuracy parity [4], predictive equality [7], and treatment equality [4]. While this strategy extends to many fairness metrics, it does not extend to all. Specifically, it does not work for fairness metrics that condition on $\hat{Y}$, such as *predictive parity* $P(Y = 1 \mid \hat{Y} = 1, Z = 0) = P(Y = 1 \mid \hat{Y} = 1, Z = 1)$ since then the Equation (14) denominators are no longer constants. However, as we later show in our Section 5 experiments, our approach is often still effective at replicating these fairness preferences even if it does not model them directly as feature expectations.

## 3.4 Resolving Y Observability Issue

Addressing group fairness in classification as an IRL problem presents one more challenge. In our approach, as described in our problem statement, we consider demonstrations as samples from $(Z, X, Y, \hat{Y})$, adhering to the fully observable MDP assumption of IRL. However, the expert classifier does not have access to $Y$, affecting the use of features involving $Y$. For example, an expert demonstrating a preference for accuracy will inevitably make errors, but IRL, with full access to $Y$, could misleadingly train a perfectly accurate classifier. This scenario would incorrectly suggest that any inaccuracy in expert decisions is intentional, potentially skewing the learned weights $w^L$. To address this, we cannot simply exclude $Y$ from `FairIRL`, as it relies on a linear program including $Y$-dependent fairness objectives and constraints. Instead, introduce a new constraint in our linear program, ensuring all $C^L$ classifier decisions are independent of $Y$:

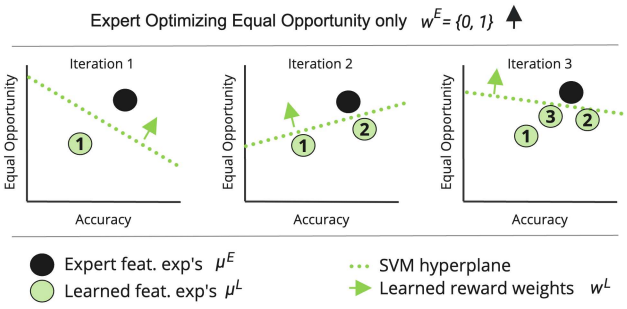
$$\lambda(z_i, x_i, y = 0, a) = \lambda(z_i, x_i, y = 1, a) \; \forall \; z, x, a \; . \tag{17}$$

This "masks" $Y$ from the `FairIRL`, preventing misattribution of rewards due to having $Y$-access when the expert did not.

# 4 ALGORITHM

In this section we propose our second primary contribution, the `FairIRL` algorithm. We begin by offering a high-level overview of `FairIRL` in Section 4.1, and then provide the technical details in Section 4.2.

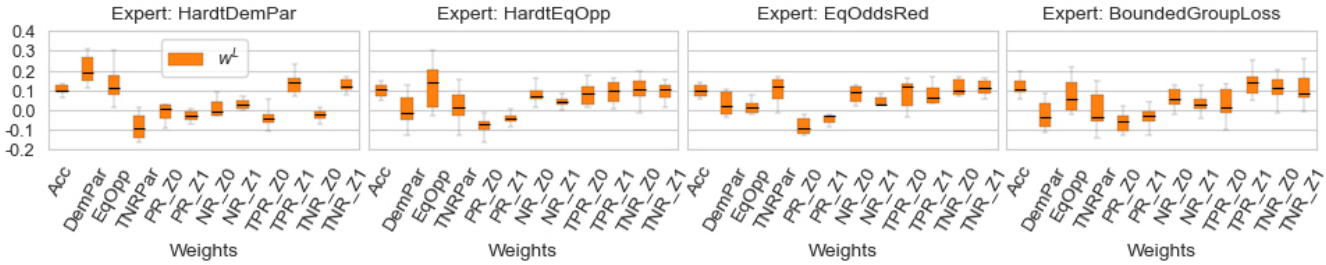
**Figure 2: An illustrative example of three iterations of `FairIRL`. We limit the visual to just two objectives to keep it two-dimensional. Notice that the green arrow, which represents the learned reward weights $w^L$, aligns closer and closer with the expert demonstrator's reward weights $w^E$ (black arrow), which faces upward.**




## 4.1 Algorithm Overview

The `FairIRL` algorithm accepts demonstrations from an expert classifier and deduces the underlying reward weights being optimized for by the expert. It then generates a new classifier that optimizes for the learned weights, applicable to either the original domain of expert demonstrations or an alternative group fairness classification dataset. Internally, `FairIRL` repeatedly learns a new classifier, and then tries to distinguish the expert's classifier from all previously learned classifiers classifier, by using an SVM to draw a hyperplane that maximizes the margin between the point represented by the expert's feature expectations and all learned feature

**Figure 3: The fairness "preference" weights learned by FairIRL across 25 trials (per expert) on the Adult dataset. The boxplots show the quartile ranges with the horizontal black line as the median.**




expectations. The unit vector orthogonal to the hyperplane represents the learned weights $w^L$, which characterizes the algorithm's best guess at the expert's reward function (fairness preferences). Therefore, each iteration adds a new learned classifier as a negative training example to the SVM, so the hyperplane in each subsequent iteration gets better and better at distinguishing the expert's feature expectations from the learned feature expectations. Figure 2 shows an illustrative example of three FairIRL iterations.

## 4.2 FairIRL

More concretely, FairIRL works as follows. FairIRL accepts three inputs: a set of expert demonstrations $(Z, X, Y, \hat{Y}^E)$ , a *feature expectation generator* FeatExpGen : $(Z, X, Y, \hat{Y}) \rightarrow \mu \in \{\mathbb{R}\}^k \; \exists k > 0$ , and a *target dataset* $(Z', X', Y')$ . and returns two outputs: a set of weights $w^L$, representing the algorithm's belief of the expert's reward function, and a new classifier $C^L$ optimal for weights $w^L$ on the target dataset. This target dataset can be either the *source dataset* from which expert demonstrations were provided, or a different dataset conforming to the group fairness classification problem outlined in Section 2.1, and where FeatExpGen is well-defined.

FairIRL builds on the max-margin IRL algorithm of Abbeel and Ng [1]. It seeks a reward function $R^L$ whose optimal classifier $C^L$ yields feature expectations within $\epsilon$ of those of the expert's demonstrations, which presumably optimize the expert's unknown reward function $R^E = w^E \cdot \mu^E$. Both their approach, and ours, reduce to seeking feature expectations $\mu^L$ within $\epsilon$ of the expert's feature expectations $\mu^E$. Algorithm 1 shows the key parts of our FairIRL algorithm. Relative to those of Abbeel and Ng, the major technical innovations are our way of computing "feature expectations" for non-linear fairness metrics (FeatExpGen) and our LP-based approach to finding an optimal classifier (LP).

The algorithm begins by processing expert demonstrations on a source dataset $(Z, X, Y, \hat{Y}^E)$, using feature expectation generator FeatExpGen. A random classifier is initialized as the first "learned" classifier. This classifier is used to generate predictions, which then further generate the first set of learned feature expectations. The core of the algorithm is the while loop, where it repeatedly fits an SVM to distinguish the expert's feature expectations $\mu^E$ from the learned feature expectations $\mu^L$ by computing the hyperplane that maximizes the margin between them (Line 10). Line 11 computes the unit vector orthogonal to the hyperplane to get the learned reward weights $\mathbf{w^L}[i]$; Line 12 computes the hyperplane margin (error); and Line 13 computes a new learned classifier $\mathbf{C^L}[i]$ by optimizing for the learned weights using the linear program defined in Sections 3.2–3.4.

Each iteration of the loop adds another learned feature expectation to $\mu^{\mathbf{L}}$ which are the negative training examples for the SVM. Therefore, with each iteration, the SVM hyperplane becomes increasingly better at distinguishing the expert's feature expectations from the learned feature expectations, and so the learned reward weights $\mathbf{w^L}$ become increasingly more similar to the expert's true reward. The iterative process continues until the error (SVM margin) falls below a pre-defined threshold $\epsilon$, indicating convergence, or if the error starts increasing. Increasing error suggests the learner is outperforming the expert on their own objectives, since this results in the feature expectations become inseparable by a linear boundary. The while loop stops when this happens since any further efforts to draw a hyperplane result in inaccurate weights. Once the algorithm converges, it selects the set of weights $(w^{L^*})$ corresponding to the iteration with the lowest error. [2] The final step involves applying the Section 3 linear program for weights $w^{L^*}$ on the target dataset, which we then use to compute final learned classifier $C^{L'}$. [3]

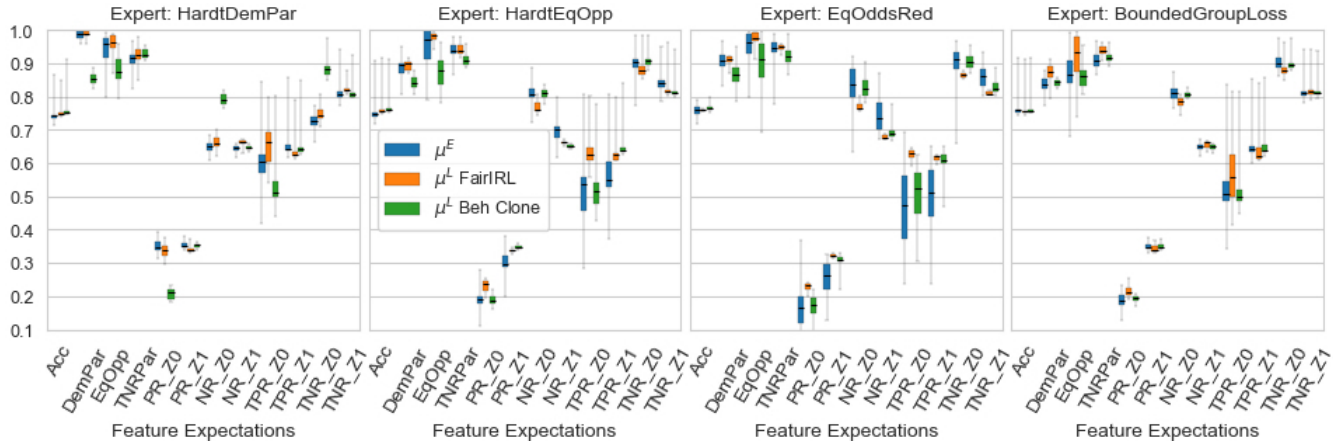
# 5 EXPERIMENTS

We aim to answer the following question: can we leverage existing fairness knowledge, embodied in algorithms or human experts, to learn fair classifiers in new domains where fairness has not been explicitly established? We decompose this into three questions, which we systematically address with a subsequent experiment:

(1) Recovery: Can we effectively capture the fairness preferences embedded within existing algorithms or expert demonstrations? This is the focus of Section 5.1.
(2) Generalization: Assuming successful recovery, are these preferences robust in that they can be used to generate new fair policies after a covariate shift? This is the focus of Section 5.2.
(3) Transfer Learning: Can we transfer these learned fairness preferences across different contexts, effectively bridging the gap between established fairness knowledge and its application in new domains? This is the focus of Section 5.3.

[2] The last iteration of the loop has the lowest error unless FairIRL was able to outperform the expert, in which case it is the second to last.
[3] Although our proposed FairIRL implementation uses the sensitive attribute $Z$ in both training and inference for simplicity, it is only needed during training for feature expectation matching. Once the weights are learned, access $Z$ is no longer required, so if desired we could omit $Z$ at inference by computing a classifier (Line 20) that does not have access to $Z$ as an input using any standard method that can optimize our weights.

**Figure 4: Observed feature expectation measures of three different classifiers on the second split of the Adult dataset. `Expert` (blue): is the original expert demonstrator. `FairIRL` (orange): the classifier computed by optimizing the weights it learned. `BehClone` (green): the classifier computed by training a model to predict the expert's predictions directly.**




For each of the 25 trials in our experiments, we train four classifiers using known fairness algorithms. These trained classifiers act as our "experts", and generate "demonstrations" by predicting on hold-out portions of the same datasets.

*Datasets.* We consider three well-studied fairness classification datasets: Adult [8], ACSIncome [9], and Boston Housing [13]. The aim of Adult is to predict an individual's annual income, while ACSIncome, a larger alternative, allows for analysis of covariate shifts due to its geographic segmentation. The goal of Boston Housing is to predict district median real estate values, which we convert to binary outcomes based on the median value. In both the Adult and ACSIncome datasets, we set race as the protected attribute, with white individuals labeled as $Z = 1$ and non-white as $Z = 0$. For Boston Housing, we also consider race as the protected attribute, and we make it binary based on whether the proportion of Black residents in a district is higher than the overall median.

*Feature Expectations and Fairness Measures.* Table 1 shows our selection of feature expectations, which are used by `FairIRL` to determine classifier similarity. We tested several combinations of features and found this set to be the most reliable across different situations. We compute these feature expectations using the method described in Section 3.3. Three of the feature expectations correspond to fairness measures (`DemPar`, `EqOpp`, `TNRPar`), but there are additional fairness measures we wish to use when evaluating how well `FairIRL` generalizes to fairness not explicitly captured as feature expectations. Table 1 also shows the full set of fairness and efficiency measures we use to evaluate the performance of our fair classifiers.

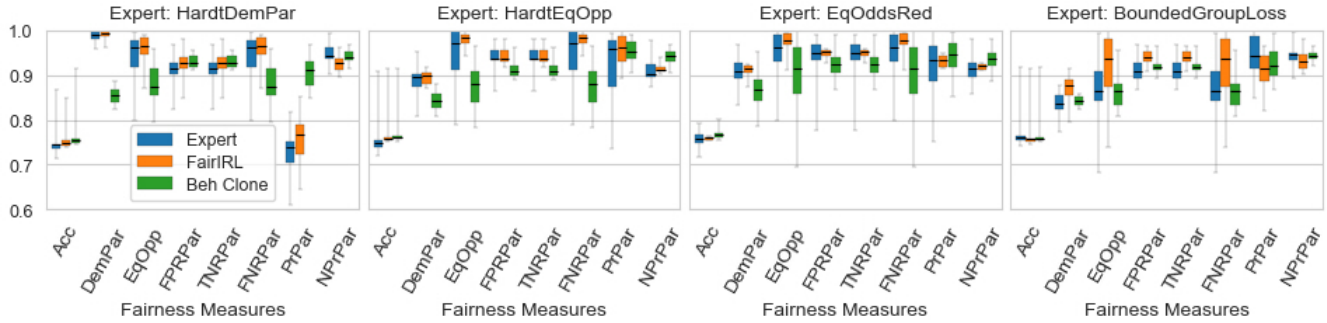
*Generating Demonstrations and Learning Classifiers.* The dataset is divided into three splits: 40% for the first split, and 30% each for the second and third splits. the first split trains the expert demonstrator classifiers. The trained experts make predictions on the second split to generate the demonstrations. These demonstrations are then used by `FairIRL` to learn the reward weights. The third split is used to evaluate both the expert and the learned classifiers. This process is repeated 25 times with different random samples of 25,000 records (pre-split) each. We utilize four distinct fairness classification algorithms as our experts. The first two, `HardtDemPar` and `HardtEqOpp` [12], are post-processing techniques optimizing for demographic parity and equal opportunity, respectively, with fairness objectives explicitly represented as feature expectations. The other two algorithms, `EqOddsRed` and `BoundedGroupLoss`, provide insights into `FairIRL`'s performance when expert preferences are not explicitly captured as feature expectations. `EqOddsRed` is a reduction technique focusing on minimizing the equalized odds difference, and `BoundedGroupLoss` aims to keep prediction errors within certain bounds for different groups. [2] All experts are generated using the Fairlearn Python package [5], with XGBoost as their estimator.

*Benchmark.* To evaluate `FairIRL`'s performance in adapting fairness preferences, we use behavioral cloning as a baseline. This method aims to replicate the expert's predictions directly by training classifiers to predict the expert's own $\hat{Y}$ predictions. These classifiers, which are XGBoost classifiers, match their experts' estimator structure. By comparing `FairIRL`'s performance to that of these cloned models, we can assess how effectively FairIRL captures the objectives of the expert, as opposed to just trying to mimic expert behaviors.

## 5.1 Recovery: Learning Expert Fairness Preferences

We begin our investigation into `FairIRL`'s ability to recover fairness preferences by visually inspecting the learned weights for any noticable patterns that align with known fairness objectives. Figure 3 shows the weights learned from four experts using the Adult dataset. Key observations include `HardtDemPar` showing a prominent `DemPar` weight, consistent with its fairness goal, and `HardtEqOpp` following suit with a high `EqOpp` weight. `EqOddsRed`, lacking an explicit feature expectation matching its objective, exhibits polarizing values for `EqOpp` and `TNRPar`. This aligns with

**Figure 5: Fairness performance on Adult dataset. `Expert` (blue): experts trained on first split, then evaluated on third split. `FairIRL` (orange): classifiers computed by optimizing weights learned from expert demos (second split), then evaluated on third split. `BehClone` (green): behavioral clone classifiers trained to predict expert on second split, then evaluated on third split.**




equalized odds being a composite of `EqOpp` and `FPRPar`, the latter being $1 - $`TNRPar` . `BoundedGroupLoss` shows a mild resemblance to `HardtEqOpp` but lacks distinct patterns. Next, we directly measure the feature expectations of classifiers trained with these weights, ensuring they match those of the expert. Fig. 4 shows the feature expectations [4] of the classifiers predicting on the third dataset split. `FairIRL` consistently outperforms behavioral cloning in replicating expert predictions, except for `BoundedGroupLoss` where the gap is marginally wider. `FairIRL`'s ability to match feature expectations on unseen data is encouraging, but it does not guarantee the ability to generalize to all fairness preferences. This is because `FairIRL` optimizes for feature matching explicitly, and not all fairness measures are captured as feature expectations. A more robust assessment of `FairIRL` involves evaluating its performance on fairness measures not directly included in feature expectations. This evaluation tests its ability to generalize fairness beyond its explicitly defined features. In Figure 5, we compare various fairness measures, both those modeled by feature expectations and others that are not. The results show that `FairIRL` classifiers align more closely with experts `HardtDemPar`, `HardtEqOpp`, and `EqOddsRed` than those learned by behavioral cloning, indicating `FairIRL`'s proficiency in understanding and applying fairness preferences to unseen data. Furthermore, we see that `FairIRL` actually *outperforms* the expert in several instances, suggesting its potential to improve upon expert algorithms. With `BoundedGroupLoss`, `FairIRL` slightly lags behind behavioral cloning in matching the expert's behavior, opting instead for a tradeoff that favors other fairness dimensions. This indicates that even when `FairIRL` diverges from the expert, it does so strategically to achieve a favorable tradeoff in other fairness dimensions.

Comparing `FairIRL`'s learned classifier fairness performance against the experts' not only validates the recovery of expert fairness preferences, but also reveals some practical potential. For instance, even if a practitioner had access to a fair expert for their domain, `FairIRL` may enable the expert to be more robust, as demonstrated by `HardtDemPar` and `HardtEqOpp` in Figure 5.

[4]To reduce new terminology, we continue to use the expression "feature expectations" even when referring to the realized values over a set of predictions.

## 5.2 Generalization: Preserving Fairness Preferences Despite Covariate Shift

Section 5.1 showed that `FairIRL` can successfully extended expert fairness preferences within the same domain. However, it would be even better if it could adapt these preferences to different domains, particularly those with shifted data distributions where fairness considerations change due to different population demographics. For instance, a classifier that is fair for one U.S. state may not be fair in others, since U.S. states have different population demographics. As an explicit example, suppose we are seek a fair classifier to predict household income for Mississippi residents, but we only have fair demonstrations on Massachusetts, a state with much higher average incomes. We model this as an experiment by training `FairIRL` on expert demonstrations on the ACSIncome Massachusetts to learn fairness preferences. We then apply those preferences to train new fair classifiers on the Mississippi ACSIncome dataset. For comparison, we also train a behavioral cloning model that mimics the expert's predictions on the Massachusetts data.

Figure 6 shows the learned weights and performance on the Mississippi dataset. `FairIRL` consistently outperforms behavioral cloning, especially when the expert prioritizes fairness over pure accuracy (e.g., `HardtDemPar`, `HardtEqOpp`). Also, as we saw in Section 5.1, it even surpasses the expert on some fairness metrics in these scenarios. For `EqOddsRed` and `BoundedGroupLoss`, `FairIRL` matches the expert and consistently outperforms behavioral cloning, except for `PrPar` and `NPrPar` where it deviates slightly more. The observed discrepancy in `PrPar` and `NPrPar` aligns with `FairIRL`'s limitations in directly representing fairness measures that condition on predicted labels $\hat{Y}$, as explained in Section 3.3.

Our findings have two key implications. First, they showcase `FairIRL`'s ability to adapt expert fairness preferences to new domains, even when faced with significant covariate shift. This holds promise for real-world applications, where data distributions typically differ across populations. Second, `FairIRL`'s ability to outperform the expert in certain scenarios suggests it can not only learn from expert guidance but also potentially improve upon it.

**Figure 6: Top: weights learned from Massachusetts (MA) demos. Bottom: fairness performance on ACSIncome Mississippi (MS) using MA-learned weights. `Expert` (blue): experts trained on MS directly. `FairIRL` (orange): weights learned from MA demos to compute classifiers on MS. `BehClone` (green): classifiers trained to predict MA demos predictions.**

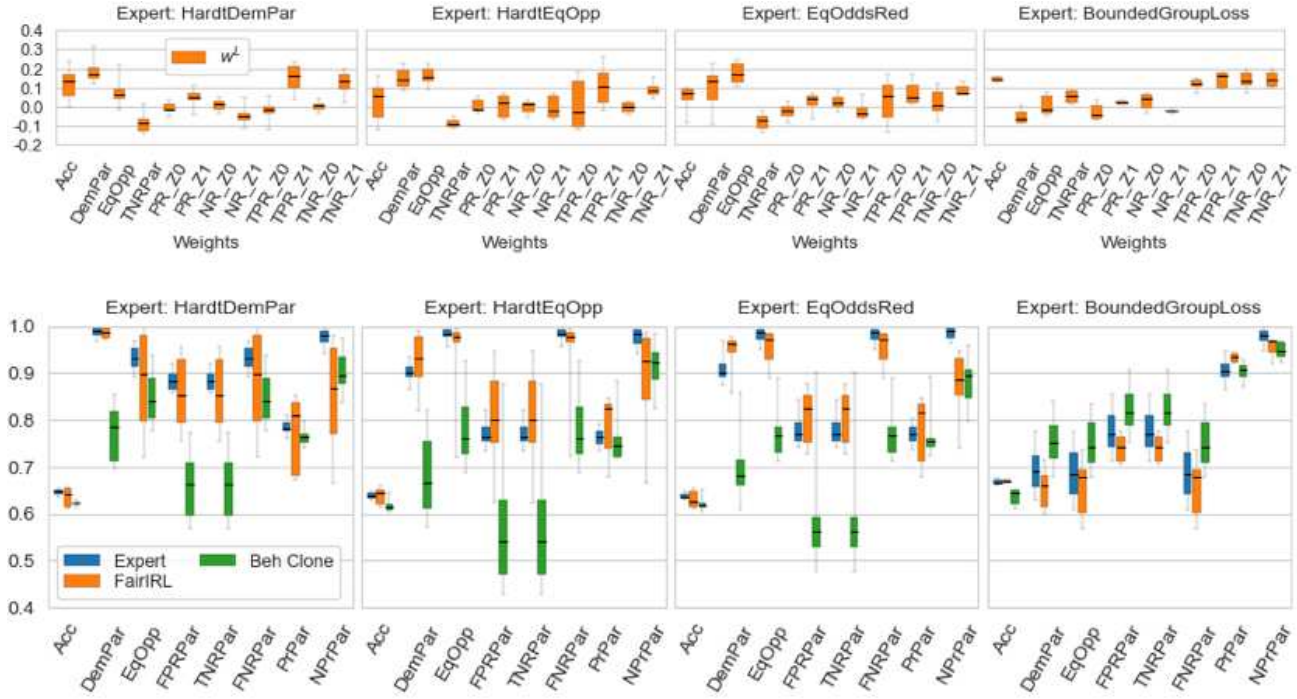
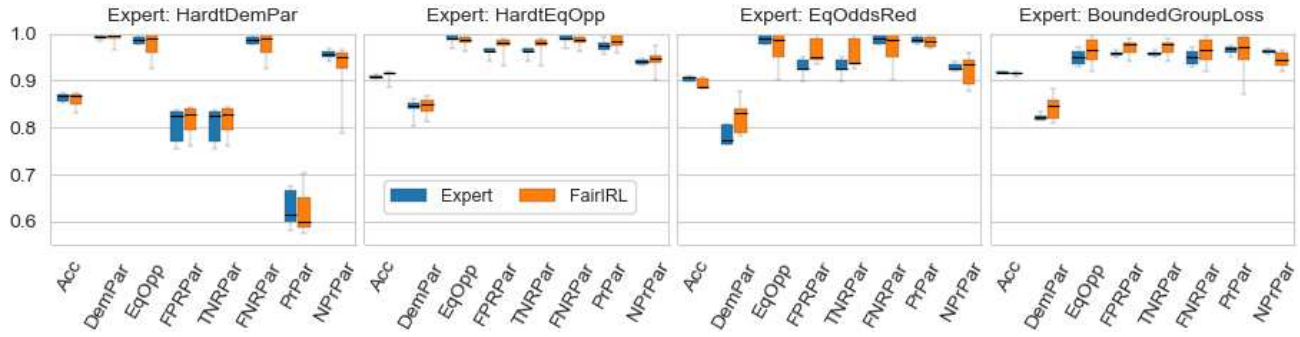


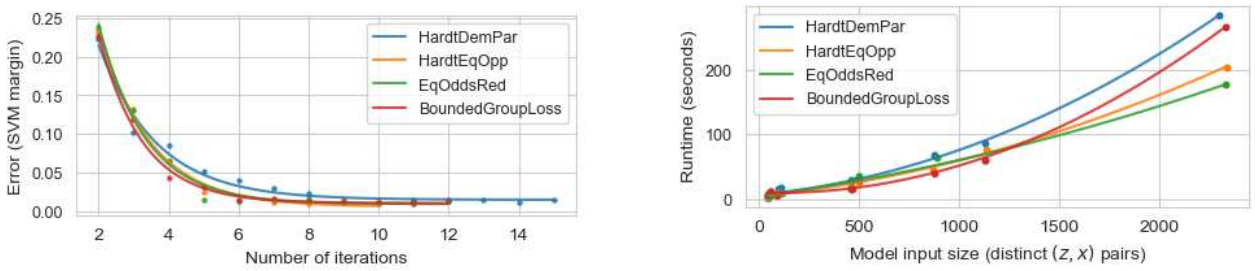

**Figure 7: Transferring fairness preferences. Classifiers computed for Boston Housing using weights learned from Adult. `Expert` (blue): expert classifiers trained *directly* on the Boston Housing dataset. `FairIRL` (orange): classifiers computed by optimizing weights learned from expert demonstrations on the Adult dataset, and then computing their optimal classifier on the Boston Housing dataset. Note that `BehClone` cannot make predictions in this context.**




## 5.3 Transfer Learning: Transferring Fairness Preferences to New Datasets

Sections 5.1 and 5.2 showed that `FairIRL`'s learned weights enabled new classifiers to extend the expert's fairness preferences to, even with covariate shift. However, a crucial question remains: can these weights generalize fairness preferences across different domains? This experiment aims to answer this question by exploring domain-invariant transfer of fairness preferences. We take the weights learned from the Adult dataset and use them to compute new classifiers on the Boston dataset, without ever showing any expert demonstrations in the target domain. Essentially, the learned weights are the sole communication channel for the expert's preferences. Figure 7 reveals the results. The `HardtDemPar` and `HardtEqOpp` `FairIRL`-learned classifiers, trained solely from Adult weights, perform remarkably similar to the expert on the target domain, despite never seeing any expert demonstrations. Notably, EqOddsRed and BoundedGroupLoss also achieve decent

**Figure 8: `FairIRL` Performance. Left: Error vs number of iterations. Right: Runtime vs input feature size.**




performance, even though their objectives aren't explicitly represented as feature expectations. This is significant because, for at least some cases, the learned weights truly represent domain-agnostic fairness preferences.

## 5.4 Performance Analysis

We conclude our experiments with a brief performance analysis. Figure 8 shows the convergence speed and runtime of `FairIRL`. The left plot shows the average error (SVM margin) at each iteration of the `FairIRL` while loop on the Adult dataset, using Section 5.1 parameters. We see that it converges quickly for all experts, reaching stable error before 15 iterations. The right plot shows the wallclock runtime of the full `FairIRL` algorithm, as as we vary the model input size (number of distinct $(z, x)$ pairs). [5] We observe a polynomial scaling of runtime with input size for each expert.

# 6 CONCLUSION

We offer a technique to frame group fairness in classification as an IRL problem, where the reward weights reflect expert fairness preferences. Our IRL-based algorithm (`FairIRL`), capable of extracting these preferences from demonstrations, trains classifiers that extend the experts' fairness preferences to unseen data on the original domain, or in new domains. Experimental results on the Adult, ACSIncome, and Boston Housing datasets validate our approach's ability to recover expert fairness preferences, learn new fair policies after covariate shifts, and apply these preferences in new domains.

We acknowledge certain limitations. First, as is the nature of IRL, if the expert demonstrator is biased, our IRL approach may also learn this bias. Second, non-identifiability in IRL, where multiple distinct reward functions can equally explain observed expert behavior, introduces complexities in ensuring the consistent application of fairness objectives when moving to new domains. Third, our method cannot represent all group fairness metrics like calibration or predictive parity, but as shown in our experiments, it can still match fairness measures it does not directly optimize.

Future work could extend in several directions. Our experiments hinted that FairIRL can sometimes surpass experts in their own fairness objectives, suggesting a future line of work to better understand how, and to what extent, `FairIRL` can be leveraged to improve a suboptimal demonstrator. Furthermore, The variability in `FairIRL`'s performance across datasets and experts suggests future research on the robustness and consistency of transferring learned fairness objectives. One might consider more advanced IRL techniques, such as Guided Cost Learning [10] which uses importance sampling to focus on demonstrations likely to be relevant in the target domain. Other IRL algorithms, such as Maximum Entropy IRL [26] or Adversarial IRL [11], could also yield more generalizable fairness preferences. Last, research into alternative optimization techniques could capture fairness metrics conditioned on predicted outcomes.

[5] All input features are treated as categorical by both the expert and `FairIRL`, including numeric ones which are binned by quantile thresholds.

# ETHICS STATEMENTS

## Ethical Considerations Statement

We carefully considered the ramifications of using potentially biased expert demonstrators. Our approach, though not perfectly capturing all fairness metrics, was transparently communicated to avoid misinterpretation. We did not involve human subjects directly in our research; hence, IRB approval was not applicable.

## Researcher Positionality Statement

We approached this research with a focus on algorithmic fairness and the application of IRL techniques. While our technical expertise guided our methodology, we recognize our perspectives could influence our interpretation of fairness and its implementation. Our collective experience in the field shapes our understanding of algorithmic bias and the necessity of fairness in machine learning.

## Adverse Impact Statement

Reflecting critically on potential adverse impacts, our work, if applied using biased demonstrators, could inadvertently perpetuate existing biases. In some special cases, the limitations in capturing all fairness metrics might lead to potentially harmful consequences as incorrect fairness preferences are learned. We emphasize the importance of ongoing evaluation and adaptation of our method to diverse contexts to mitigate these risks. Future work should focus on expanding the model's capabilities and exploring mitigation strategies for unintended consequences.

# ACKNOWLEDGMENTS

This material is based upon work supported by the NSF Program on Fairness in AI in Collaboration with Amazon under Award No. 1939743. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or Amazon.

# REFERENCES

[1] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*. 1.

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International conference on machine learning*. PMLR, 60–69.

[3] Saurabh Arora and Prashant Doshi. 2021. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence* 297 (2021), 103500.

[4] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2018. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research* (2018), 0049124118782533.

[5] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. 2020. *Fairlearn: A toolkit for assessing and improving fairness in AI.* Technical Report MSR-TR-2020-32. Microsoft. https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/

[6] Li Chen and Pearl Pu. 2004. *Survey of preference elicitation methods.* Technical Report.

[7] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.

[8] Dua Dheeru and E Karra Taniskidou. 2017. UCI machine learning repository. (2017).

[9] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems* 34 (2021), 6478–6490.

[10] Chelsea Finn, Sergey Levine, and Pieter Abbeel. 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*. PMLR, 49–58.

[11] Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248* (2017).

[12] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf

[13] David Harrison Jr and Daniel L Rubinfeld. 1978. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management* 5, 1 (1978), 81–102.

[14] Gaurush Hiranandani, Jatin Mathur, Harikrishna Narasimhan, and Oluwasanmi Koyejo. 2022. Quadratic metric elicitation for fairness and beyond. In *Uncertainty in Artificial Intelligence*. PMLR, 811–821.

[15] Gaurush Hiranandani, Harikrishna Narasimhan, and Sanmi Koyejo. 2020. Fair performance metric elicitation. *Advances in Neural Information Processing Systems* 33 (2020), 11083–11095.

[16] Zhimeng Jiang, Xiaotian Han, Hongye Jin, Guanchu Wang, Na Zou, and Xia Hu. 2023. Weight perturbation can help fairness under distribution shift. *arXiv preprint arXiv:2303.03300* (2023).

[17] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807* (2016).

[18] Sergey Levine, Zoran Popovic, and Vladlen Koltun. 2011. Nonlinear inverse reinforcement learning with gaussian processes. *Advances in neural information processing systems* 24 (2011).

[19] Michael Madaio, Lisa Egede, Hariharan Subramonyam, Jennifer Wortman Vaughan, and Hanna Wallach. 2022. Assessing the Fairness of AI Systems: AI Practitioners' Processes, Challenges, and Needs for Support. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (2022), 1–26.

[20] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 2018. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*. PMLR, 3384–3393.

[21] Omid Memarrast, Linh Vu, and Brian Ziebart. 2023. Superhuman Fairness. *arXiv preprint arXiv:2301.13420* (2023).

[22] Luca Oneto, Michele Donini, Massimiliano Pontil, and Andreas Maurer. 2020. Learning fair and transferable representations with theoretical guarantees. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 30–39.

[23] Ashkan Rezaei, Anqi Liu, Omid Memarrast, and Brian D Ziebart. 2021. Robust fairness under covariate shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9419–9427.

[24] Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference* 90, 2 (2000), 227–244.

[25] Brian Ziebart, Sanjiban Choudhury, Xinyan Yan, and Paul Vernaza. 2022. Towards Uniformly Superhuman Autonomy via Subdominance Minimization. In *International Conference on Machine Learning*. PMLR, 27654–27670.

[26] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.