IS FAIRNESS ONLY METRIC DEEP? EVALUATING AND ADDRESSING SUBGROUP GAPS IN DML

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep metric learning (DML) enables learning with less supervision through its emphasis on the similarity structure of representations. There has been much work on improving generalization of DML in settings like zero-shot retrieval, but little is known about its implications for fairness. In this paper, we are the first to evaluate state-of-the-art DML methods trained on imbalanced data, and to show the negative impact these representations have on minority subgroup performance when used for downstream tasks. In this work, we first define fairness in DML through an analysis of three properties of the representation space – inter-class alignment, intra-class alignment, and uniformity – and propose ***finDML***, the ***f***airness ***i***n ***n***on-balanced ***DML*** benchmark to characterize representation fairness. Utilizing *finDML*, we find bias in DML representations to propagate to common downstream classification tasks. Surprisingly, this bias is propagated even when training data in the downstream task is re-balanced. To address this problem, we present Partial Attribute De-correlation (***PARADE***) to de-correlate feature representations from sensitive attributes and reduce performance gaps between subgroups in both embedding space and downstream metrics.

1 INTRODUCTION

Deep metric learning (DML) extends standard metric learning to deep neural networks, where the goal is to learn metric spaces such that embedded data sample distance is connected to actual semantic similarities (Globerson & Roweis, 2006; Weinberger et al., 2006; Hoffer & Ailon, 2018; Wang et al., 2014). The explicit optimization of similarity makes deep metric spaces well suited for usage in unseen classes, such as zero-shot image or video retrieval or facial re-identification (Milbich et al., 2021; Roth et al., 2020c; Musgrave et al., 2020; Hoffer & Ailon, 2018; Wang et al., 2014; Schroff et al., 2015; Wu et al., 2018; Roth et al., 2020c; Brattoli et al., 2020; Hu et al., 2014; Deng et al., 2019; Liu et al., 2017). However, while DML is effective in establishing notions of similarity, work describing potential fairness issues is limited to individual fairness in standard metric learning (Ilvento, 2020), disregarding embedding models.

Indeed, the impacts and metrics of fairness are well studied in machine learning (ML) generally, and representation learning specifically (Dwork et al., 2012; Mehrabi et al., 2019; Locatello et al., 2019b). This is especially true on high-risk tasks such as facial recognition and judicial decision-making (Chouldechova, 2017; Berk, 2017), where there are known risks to minoritized subgroups (Samadi et al., 2018). Yet, relatively little work has been done in the domain of DML (Rosenberg et al., 2021). It is crucial to address this knowledge gap – if DML embeddings are used to create *upstream* embeddings that facilitate *downstream* transfer tasks, biases may propagate unknowingly.

To tackle this issue, this work first proposes a benchmark to characterize ***f***airness ***i***n ***n***on-balanced ***DML*** - ***finDML***. *finDML* introduces three subgroup fairness definitions based on feature space performance metrics – recall@k, alignment and group uniformity. These metrics measure clustering ability and generalization performance via feature space uniformity. Thus, we select the metrics for our definitions to enforce independence between inclusion in a particular cluster or class, and a protected attribute (given the ground-truth label). We leverage existing datasets with fairness limitations (CelebA (Liu et al., 2015) and LFW (Huang et al., 2007)) and induce imbalance in training data of standard DML benchmarks, CARS196 (Krause et al., 2013) and CUB200 (Wah et al., 2011), in order to create an effective benchmark for fairness analysis in DML.

Making use of *finDML*, we then perform an evaluation of 11 state-of-the-art (SOTA) DML methods representing frequently used losses and sampling strategies, including: ranking-based losses (Wang et al., 2014; Hoffer & Ailon, 2018), proxy-based (Kim et al., 2020) losses, semi-hard sampling (Schroff et al., 2015) and distance-weighted sampling (Wu et al., 2018). Our experiments suggest that imbalanced data during upstream embedding impacts the fairness of all benchmarks methods in both upstream embeddings (subgroup gaps up to 21%) as well as downstream classifications (subgroup gaps up to 45.9%). **This imbalance is significant even when downstream classifiers are given access to balanced training data, indicating that data cannot naively be used to de-bias downstream classifiers from imbalanced embeddings.**

Finally, inspired by prior work in DML on multi-feature learning (Milbich et al., 2020), we introduce PARtial Attribute DE-correlation (PARADE). PARADE addresses imbalance by de-correlating two learned embeddings: one learnt to represent similarity in class labels, and one learnt to represent similarity in the values of a sensitive attribute, which is discarded at test-time. This creates a model in which the ultimate target class embeddings have been de-correlated from the sensitive attributes of the input. We note that as opposed to previous work on variational latent spaces, PARADE de-correlates a learned similarity metric. We find that PARADE reduces gaps of SOTA DML methods by up to 2% downstream in *finDML*.

In total, our contributions can be summarized as follows:

1. We define ***finDML***; introducing three definitions of fairness in DML to capture multi-faceted minoritized subgroup performance in upstream embeddings through focus on feature representation characteristics across subgroups, and five datasets for benchmarking.
2. We analyze SOTA DML methods using *finDML*, and find that common DML approaches are significantly impacted by imbalanced data. We show empirically that learned embedding bias cannot be overcome by naive inclusion of balanced data in downstream classifiers.
3. We present ***PARADE***, a novel a novel adaptation of previous zero-shot generalization techniques to enhance fairness guarantees through de-correlation of class discriminative features with sensitive attributes.

2 BACKGROUND

Deep Metric Learning DML extends standard metric learning by fusing feature extraction and learning a parametrized metric space into one end-to-end learnable setup. In this setting, a large convolutional network ψ provides the mapping to a feature space Ψ, while a small network f, usually a single linear layer, generates the final mapping to the metric or embedding space Φ. The overall mapping from the image space X is thus given by $\phi = f \circ \psi$. Generally, the embedding space is projected on the unit hypersphere $\mathcal{S}^{D-1}$ through normalization (Weisstein, 2002; Wu et al., 2018; Roth et al., 2020c; Wang & Isola, 2020) to limit the volume of the representation space with increasing embedding dimensionality. The embedding network ϕ is then trained to provide a metric space Φ that operates well under some predefined, usually non-parametric metric such as the Euclidean or cosine distance defined over Φ.

Typical objectives used to learn such metric spaces range from contrastive ranking-based training using tuples of data, such as pairwise (Hadsell et al., 2006), triplet- (Schroff et al., 2015; Wu et al., 2018) or higher-order tuple-based training (Sohn, 2016; Wang et al., 2020a), procedures to bring down the effective complexity of the tuple space (Schroff et al., 2015; Harwood et al., 2017; Wu et al., 2018) or the introduction of learnable tuple constituents (Movshovitz-Attias et al., 2017; Qian et al., 2019; Kim et al., 2020).

More recent work (Milbich et al., 2020; Roth et al., 2020c; Jacob et al., 2019) extends standard DML training through incorporation of objectives going beyond just sole class label discrimination: e.g., through the introduction of artificial samples (Lin et al., 2018; Duan et al., 2018), regularization of higher-order moments (Jacob et al., 2019), curriculum learning (Zheng et al., 2019; Harwood et al., 2017; Roth et al., 2020a), knowledge distillation (Roth et al., 2020b) or the inclusion of additional features (DiVA) to produce diverse and de-correlated representations (Milbich et al., 2020).

DML Evaluation Standard performance measures reflect the goal of DML: namely, optimizing an embedding space Φ for best transfer to new test classes via learning semantic similarities. As

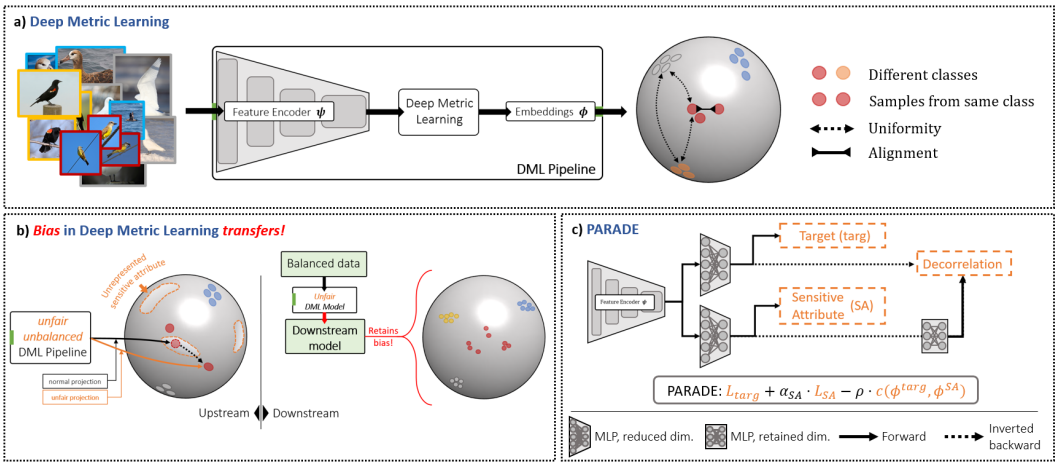


Figure 1: **a)** Visualization of the standard DML pipelines and the aspects of intra-class alignment and uniformity in the embedding space. **b)** Infographic of the fairness issue in DML, where learned representational bias can even transfer to downstream models building on previously learned representations. **c)** Layout of our proposed PARADE approach to better incorporate sensitive attribute context and improve representational fairness.

immediate applications are commonly found in zero-shot clustering or image retrieval, respective retrieval and clustering metrics are predominantly utilized for evaluation. Recall@k (Jegou et al., 2011) or mean average precision measured on recall (Roth et al., 2020c; Musgrave et al., 2020) typically estimate retrieval performance. Normalized mutual information (NMI) on clustered embeddings (Manning et al., 2010) is used as a proxy for clustering quality (see Supplemental for detailed definitions). We leverage these performance metrics to inform *finDML* and our experiments.

Fairness in Classification Formalizing fairness in ML continues to be an open problem (Mehrabi et al., 2019; Chen et al., 2018a; Chouldechova, 2017; Berk, 2017; Locatello et al., 2019b; Chouldechova & Roth, 2018; Dwork et al., 2012; Hardt et al., 2016; Zafar et al., 2017). In classification, definitions for fairness such as demographic parity, equalized odds, and equality of opportunity, rely on model outputs across the random variables of protected attribute and ground-truth label (Dwork et al., 2012; Hardt et al., 2016).

Fairness in Representations A more relevant family of fairness definitions for DML would be those explored in fairness for general representation learning (Edwards & Storkey, 2015; Beutel et al., 2017; Louizos et al., 2015; Madras et al., 2018). Here, the goal is to learn a *fair* mapping from an original domain to a latent domain so that classifiers trained on these representations are more likely to be agnostic to the sensitive attribute in unknown downstream tasks. This assumption distinguishes our setting from previous fairness work in which the downstream tasks are known at train time (Madras et al., 2018; Edwards & Storkey, 2015; Moyer et al., 2018; Song et al., 2019; Jaiswal et al., 2019). DML differs from this form of representation learning as it aims to learn a mapping capturing semantic similarity, as opposed to latent space representation.

Earlier works in fair representation learning intended to obfuscate *any* information about sensitive attributes to approximately satisfy demographic parity (Zemel et al., 2013) while a wealth of more recent works focus on using adversarial methods or feature disentanglement in latent spaces of VAEs (Locatello et al., 2019a; Kingma & Welling, 2013; Gretton et al., 2006; Louizos et al., 2015; Amini et al., 2019; Alemi et al., 2018; Burgess et al., 2018; Chen et al., 2018b; Kim & Mnih, 2018; Esmaeili et al., 2019; Song et al., 2019; Gitiaux & Rangwala, 2021; Rodríguez-Gálvez et al., 2020; Sarhan et al., 2020; Paul & Burlina, 2021; Chakraborty et al., 2020). In this setting, the literature has focused on optimizing on approximations of the mutual information between representations and sensitive attributes: maximum mean discrepancy (Gretton et al., 2006) for deterministic or variational (Li et al., 2014; Louizos et al., 2015) autoencoders (VAEs); cross-entropy of an adversarial network that predicts sensitive attributes from the representations (Edwards & Storkey, 2015; Xie et al., 2017; Beutel et al., 2017; Zhang et al., 2018; Madras et al., 2018; Adel et al., 2019; Zhao &

Gordon, 2019; Xu et al., 2018); balanced error rate on both target loss and adversary loss (Zhao et al., 2019); Weak-Conditional InfoNCE for conditional contrastive learning (Tsai et al., 2021).

PARADE shares aspects of these previous methods in its choice of de-correlation or disentanglement. However, PARADE de-correlates the learned similarity metric as opposed to the latent space. In addition, with *DML-specific* criteria, PARADE learns similarities *over the sensitive attribute* while not directly removing *all* information about the sensitive attribute, as the sensitive attribute and target class embeddings share a base network.

3 EXTENDING FAIRNESS TO DML - *finDML* BENCHMARK

To characterize fairness with *finDML*, this section introduces the key constituents – definitions to characterize fairness in embedding spaces and respective benchmark datasets.

3.1 PRELIMINARIES

Our embedding space fairness definitions rely on embedding space metrics adapted from (Wang & Isola, 2020) and (Roth et al., 2020c), namely alignment and uniformity. Both metrics we use to characterize embeddings for our definitions in the next section (*intra-* as well as *inter-class alignment* and *uniformity*) have been successfully linked to generalization performance in contrastive self-supervised and metric learning models (Wang & Isola, 2020; Roth et al., 2020c; Sinha et al., 2020). Alignment succinctly captures the similarity structure learned by the representation space with respect to the target labels through measuring distances between pairs of samples. On the other hand, notions of uniformity can differ. Uniformity of the *sample distribution over the hypersphere* has been studied through the radial basis function (RBF) over pairs of samples. Alternatively, uniformity of the *feature space* has been studied through the KL-divergence $\mathcal{D}_{\text{KL}}$ between the discrete uniform distribution $\mathcal{U}_D$ and the sorted singular value distribution $\mathcal{S}_{\phi(X)}$ of the representation space ϕ on dataset X.

$$U_{\text{KL}}(X) = \mathcal{D}_{\text{KL}}\left(\mathcal{U}_{\text{D}}, \mathcal{S}_{\phi(X)}\right) \tag{1}$$

Here, lower scores indicate more significant directions of variance in learned representations. Both introduced notions of uniformity represent important aspects of the embedding space, but the computational overhead in computing RBF over all pairs of samples in large datasets makes it impractical for our uses and is less interpretable than U_{KL}. Therefore, we leave the uniformity metric utilized in *finDML* general, but utilize U_{KL} for our experiments.

3.2 DEFINING FAIRNESS

Building on the aforementioned performance metrics, we introduce three definitions for fairness in the embedding spaces of DML models. As the recall@k and alignment metrics inform inclusion in an embedded cluster (or class), we follow fair classification literature in the motivation for our first fairness definition: inclusion in a class should be independent of a protected attribute *given* the ground-truth label. Thus, we examine the probability of encountering a data instance of the same class in a data point's k-nearest neighbors to form the first definition. The second definition relies on equal expectation of alignment across sensitive attribute values. Departing from classification literature, our third definition encapsulates fairness in a task-agnostic sense (as DML is often applied in such settings): fairness across the "goodness" of the learned features via a uniformity metric.

Let X denote the input data, and A a protected attribute variable. Denote X_a the partition of X with attribute $a \in A$. To recap common DML terminology, a *positive* pair of samples is defined as $(x_1, x_2) \in X \times X$ s.t. the class label of x_1 and x_2 are identical. A *negative* pair of samples is defined as $(x_1, x_2) \in X \times X$ such that the class label of x_1 and x_2 differ. Let $\mathbb{P}_a$ denote the set of all *positive pairs* s.t. at least one of x_1 or x_2 has attribute $a \in A$, and analogously for $\mathbb{N}_a$ and *negative pairs*.

Definition 1 (K-Close Fairness)**.** *Define $NN_k : \Phi \subset \mathcal{S}^{D-1} \rightarrow \mathcal{P}(X)$ as a function that receives a point $\phi(x) \in \Phi$ and returns a set in the powerset of X, $\mathcal{P}(X)$, containing points in X that map to the k nearest neighbors of $\phi(x)$ in Φ. Thus, ϕ is k-close fair with respect to attribute A if:*

$$\Pr_{x \in X_a}(\exists \tilde{x} \in NN_k(\phi(x)) s.t. Y(\tilde{x}) = Y(x)) = \Pr_{x \in X_b}(\exists \tilde{x} \in NN_k(\phi(x)) s.t. Y(\tilde{x}) = Y(x)) \quad \forall a, b \in A \tag{2}$$

Note: the criteria weakens as k increases, similar to recall@k.

Definition 2 (Alignment). *ϕ is fair, according to alignment with respect to attribute A, if:*

$$\mathbb{E}_{(x_1,x_2)\in\mathbb{P}_a}[||\phi(x_1)-\phi(x_2)||^2] = \mathbb{E}_{(x_1,x_2)\in\mathbb{P}_b}[||\phi(x_1)-\phi(x_2)||^2] \tag{3}$$

$$\mathbb{E}_{(x_1,x_2)\in\mathbb{N}_a}[||\phi(x_1)-\phi(x_2)||^2] = \mathbb{E}_{(x_1,x_2)\in\mathbb{N}_b}[||\phi(x_1)-\phi(x_2)||^2] \quad \forall a,b\in A \tag{4}$$

i.e. the expectation of the alignment is equal across domain of A.

Definition 3 (Uniformity Across Groups). *ϕ is fair, according to uniformity, and with respect to attribute A, if the expectation of the uniformity is equal across domain of A:*

$$U(\phi(X_a)) = U(\phi(X_b)) \quad \forall a,b\in A \tag{5}$$

where $U(\cdot)$ denotes some measure of uniformity over a set $V\in\mathcal{S}^{D-1}$.

3.3 CONSTRUCTED *finDML* BENCHMARK DATASETS

finDML encompasses existing DML benchmark datasets, CUB200 and CARS196, and facial recognition datasets, CelebA and LFW (Wah et al., 2011; Krause et al., 2013; Liu et al., 2015; Huang et al., 2007). For fairness analysis, we investigate bird color in CUB200[1], Race in LFW and Skintone in CelebA (Kumar et al., 2009). A detailed description of dataset and attribute labeling is included in the Supplemental. To create additional *fairness* benchmarks, we induce class imbalance in CUB200 and CARS196, as both datasets are naturally balanced w.r.t. class.

Manually Introduced Class Imbalance We introduce imbalance by reducing the number of training data samples of 50 randomly selected classes by 90% (Imbalanced). We run an experiment with the original datasets as a balanced control (Balanced) for comparison. In the imbalanced setting, we adjust (increase) the number of training samples of the majoritized groups to match the number of datapoints in the balanced control experiments. We average metrics over 10 sets of 50 randomly selected classes for imbalanced experiments. We use the standard ratio of $50-50$ for train-test split of these datasets, but split over number of data points per class, as opposed to splitting over the classes themselves. The manually imbalanced datasets are used to benchmark standard DML methods, validate our framework, and analyze downstream effects.

Although dataset imbalance does not constitute the sole source of bias in machine learning applications, unfairness as a result of imbalance is the most well-understood in the literature (Chen et al., 2018a). Additionally, we do not assume for our naturally imbalanced datasets, particularly the facial datasets, that attribute imbalance is the *only* source of bias we observe.

4 PARTIAL ATTRIBUTE DE-CORRELATION (PARADE)

In this section, we present Partial Attribute De-correlation, or PARADE, in which we incorporate adversarial separation (Milbich et al., 2020) during training to de-correlate separate embeddings. We enumerate several significant changes: 1) only target embedding released at test-time; 2) triplet formation and loss term w.r.t. sensitive attribute; 3) de-correlation with sensitive attribute as opposed to de-correlation to reduce redundancy in concatenated feature space. These two representations branch off from the deep metric embedding model at the last layer. The two representations encode the similarity metrics learned over the sensitive attribute and target class, respectively. The sensitive attribute embedding layer is discarded at test time. The resulting network expresses a similarity metric with respect to the target class, de-correlated from the sensitive attribute (Figure 1). Therefore, PARADE figuratively optimizes the first two fairness definitions proposed in Section 3.2 via an objective that maximizes independence between the sensitive attribute and target class.

Objective Term Per Embedding To achieve efficient training and de-correlation of the *target class* and the *sensitive attribute* embedding layers, we simultaneously train both layers that branch from the penultimate layer of the model and de-correlate at each iteration. Because PARADE must learn one embedding w.r.t. target class (ϕ_{targ}) and one embedding w.r.t. the sensitive attribute (ϕ_{SA}), we introduce separate objectives for each embedding:

$$\mathcal{L}_{targ} = \frac{1}{N}\sum_{t\sim\mathcal{T}_{targ}}\mathcal{L}(t) \qquad \mathcal{L}_{SA} = \frac{1}{N}\sum_{t\sim\mathcal{T}_{SA}}\mathcal{L}(t)$$

[1]While bird color in CUB200 does not represent a real-world fairness setting, CUB200 is widely used as a DML benchmark. Thus, a fairness angle allows fairness analysis of previous methods benchmarked on CUB200.

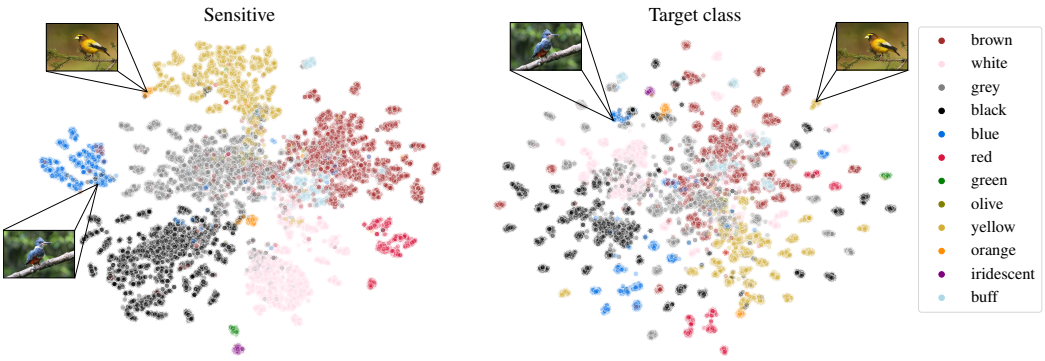


Figure 2: A t-SNE (Maaten & Hinton, 2008) visualization of the two distinct PARADE embeddings for bird color CUB200 experiments: the sensitive attribute embedding (**left**) and the class label embedding (**right**). In the sensitive attribute embedding, both example images are mapped to clusters with birds of the same plumage (yellow and blue, respectively). Due to de-correlation, in the class label embedding, the images are separated from the region of space with other birds of the same plumage, but are still well-clustered, indicating that PARADE can find other attributes to distinguish these species clusters.

where N is the number of training triplet samples, and $\mathcal{L}$ represents a generic loss function, such as triplet loss (Hoffer & Ailon, 2018). We use $t \sim \mathcal{T}_{targ}$ to illustrate sampling over triplets of the form (x_a, x_p, x_n) where x_a and x_p are of the same target class and x_a and x_n are of differing target classes. Similarly, $t \sim \mathcal{T}_{SA}$ indicates sampling over triplets of the form (x_a, x_p, x_n) where x_a and x_p are of the same sensitive attribute subgroup and x_a and x_n are of differing sensitive attribute subgroups. See Figure 2 for a t-SNE visualization of the distinct embeddings of PARADE.

Partial De-correlation In order to minimize the correlation between ϕ^{targ} and ϕ^{SA}, we use the adversarial separation (de-correlation) method from (Milbich et al., 2020), which minimizes the mutual information between a pair of embeddings. The task of mutual information minimization is accomplished through learning an MLP to maximize the pair's correlation, c, and consequently performing a gradient reversal R, which inverts the gradients during backpropagation. The MLP, ξ, is trained to maximize $c(\phi_i^{targ}, \phi_i^{SA}) = \|R(\phi_i^{targ}) \bigodot \xi(R(\phi_i^{SA}))\|_2^2$, s.t. $\bigodot$ denotes element-wise multiplication. Combining the loss terms results in total loss:

$$\mathcal{L}_{PARADE} = \mathcal{L}_{targ} + \alpha_{SA}\mathcal{L}_{SA} - \rho \cdot c(\phi^{targ}, \phi^{SA})$$

where α_{SA} weights the sensitive attribute loss and ρ weights the degree of de-correlation. ρ modulates the de-correlation term to allow ψ to retain some attribute information (i.e. *partial* de-correlation). Thus, the deployed model $\phi_{targ} = f_{targ} \circ \psi$ can retain information about the sensitive attribute in its feature representations, as $\alpha_{SA}\mathcal{L}_{SA}$ appears in the loss function back-propagated through the full model ψ. The extent to which the sensitive attribute affects the output features is controlled by α_{SA}; we suggest optimizing $\alpha_{SA} \in (0, 1)$ and ρ through maximization of worst-group performance (Lahoti et al., 2020) (See Supplemental C.5 for further analysis of PARADE hyperparameters).

5 Experiments

Baseline DML Methods For all datasets, we use a ResNet-50 (He et al., 2016) architecture with best performing parameters on a validation set (for further implementation details, see Supplemental). To investigate a sweeping set of frequently used DML methods, we benchmark across a diverse, representative set of 11 techniques, including: three standard ranking-based losses (margin, triplet, n-pair, and contrastive) three batch mining strategies (random, semi-hard and distance-weighted sampling) and three common loss functions (multisimilarity loss, ArcFace loss, for handling facial datasets, and proxy-based loss, ProxyNCA) (Hoffer & Ailon, 2018; Hadsell et al., 2006; Wu et al., 2018; Sohn, 2016; Hadsell et al., 2006; Kim et al., 2020; Wang et al., 2020a; Deng et al., 2019; Wu et al., 2018; Schroff et al., 2015). See Supplementary for more details.

Fairness Evaluation In the embedding space, we analyze fairness via performance gaps between minoritized groups and majoritized groups, or worst-group performance gaps (Lahoti et al., 2020). For fairness of the feature representations, we compute gaps in three metrics: recall@k and NMI for intra- and inter-class distance (Section 3.2), and the uniformity measure U_{KL} corresponding to Definition 3 (defined in Section 3.1).

Training and Evaluation on Downstream Classifiers To link fairness performance in the embedding space to downstream classification (in which more extensive prior work has been completed), we train downstream classifiers and evaluate classification bias. After training the DML model with the aforementioned criteria, the network is fixed. The output embeddings from the image training datasets, in addition to the class labels, are used to train four downstream classification models: logistic regression (LR), support vector machine (SVM), K-Means (KM), and random forest (RF) (Pedregosa et al., 2011). In the manually imbalanced upstream setting, we train downstream classifiers on the *original balanced image datasets* to ascertain if bias incurred in the embedding can propagate downstream even if the downstream classifier is trained with real balanced data.

We execute class imbalanced experiments for CARS196 and CUB200 and vary the level of imbalance between minoritized and majoritized classes in the upstream training set.

PARADE Configuration We test Partial Attribute De-correlation, PARADE, by training models in the listed settings: manually color imbalanced dataset for CUB200, CelebA and LFW. The attribute used to train the sensitive attribute embedding for each dataset, and the attribute used for fairness evaluation. We compare PARADE with margin loss and distance-weighted sampling (Wu et al., 2018) to standard margin loss and distance-weighted sampling.

6 RESULTS

6.1 SOTA DML METHODS HAVE LARGE FAIRNESS GAPS IN *finDML* BENCHMARK

Our experiments indicate that current DML methods encounter crucial fairness limitations in the presence of imbalanced training data. Table 1 (along with a corresponding table for CARS196 in the Supplemental) demonstrate that gaps in the manually class imbalanced setting are greater than the balanced control setting. In four combinations of loss functions and sampling strategies, we do not observe a scenario in which the class imbalanced setting achieves a smaller gap than the control in the embedding space, *nor* the downstream classification. This is particularly significant due to the nature of sampling strategies studied (Wu et al., 2018; Schroff et al., 2015), which batch samples to force the model to correct “hard" examples. The results validate *finDML* as a benchmark and framework for fairness through the lens of well-studied fairness characterization in classification.

Interestingly, Table 1 displays non-negligible gaps in downstream performance metrics recall and precision even in the balanced control case. This could represent stenography of underlying structures in the data, such as car color or bird size. More likely, however, these gaps are due to use of *macro*-averaging in recall and precision calculations. Nonetheless, the manually class imbalanced settings consistently produce larger gaps.

6.2 PROPAGATION OF BIAS TO DOWNSTREAM TASKS

The tabular results emphasize a significant result: naive re-balancing with real data downstream cannot overcome bias incurred in the upstream embedding in any setting studied. Indeed, Table 1 exhibits propagation of bias from upstream embeddings (trained on imbalanced data) to downstream tasks (trained on fixed upstream embeddings with a re-balanced dataset). To provide additional context for the result, we direct to increasing use of DML models as components of larger classification models. This trend is arising in literature such as supervised contrastive learning, and recent developments in pre-training and lifting DML models for classification (Khosla et al., 2020). This necessitates tackling bias in the representation space of DML as opposed to patches downstream, and emphasizes the importance of defining fairness in this setting as done in our work.

Impact of imbalance degree on lack of fairness Figure 3 shows that gaps in downstream classification mimic those upstream, even as we vary the level of imbalance introduced when training the upstream embedding. Here, the random forest classifier sees greater gaps in downstream metrics

Table 1: *Gap study on CUB200-2011.* Average gaps in representation space and downstream classification (logistic regressor) over 10 seeds between minoritized and majoritized classes in manually class imbalanced experiments (Imbalanced) and control experiments (Balanced) for CUB200-2011. Results for CARS196 are available in the supplementary with similar conclusions. **Bold** represents larger gap for each method shown (Loss · Batch Mining).

Experiments →		Balanced	Imbalanced	Balanced	Imbalanced
Objective →		Margin · Distance		Margin · Semi-hard	
Upstream Embedding	Recall@1	0.017 ± 0.007	$\mathbf{0.212 \pm 0.029}$	0.02 ± 0.007	$\mathbf{0.187 \pm 0.031}$
	NMI	-0.001 ± 0.004	$\mathbf{0.112 \pm 0.012}$	-0.004 ± 0.004	$\mathbf{0.092 \pm 0.017}$
	U_{KL}	-0.042 ± 0.003	$\mathbf{0.0 \pm 0.002}$	-0.048 ± 0.004	$\mathbf{0.002 \pm 0.004}$
Downstream Classification	Precision	0.339 ± 0.007	$\mathbf{0.39 \pm 0.014}$	0.33 ± 0.004	$\mathbf{0.393 \pm 0.015}$
	Recall	0.36 ± 0.007	$\mathbf{0.424 \pm 0.018}$	0.351 ± 0.005	$\mathbf{0.43 \pm 0.016}$
	Accuracy	0.014 ± 0.002	$\mathbf{0.131 \pm 0.027}$	0.016 ± 0.005	$\mathbf{0.131 \pm 0.031}$
Objective →		Triplet · Distance		Triplet · Semi-hard	
Upstream Embedding	Recall@1	0.019 ± 0.006	$\mathbf{0.159 \pm 0.031}$	0.019 ± 0.006	$\mathbf{0.168 \pm 0.036}$
	NMI	-0.001 ± 0.004	$\mathbf{0.103 \pm 0.016}$	-0.004 ± 0.006	$\mathbf{0.082 \pm 0.016}$
	U_{KL}	-0.054 ± 0.006	$\mathbf{-0.004 \pm 0.009}$	-0.051 ± 0.006	$\mathbf{0.014 \pm 0.011}$
Downstream Classification	Precision	0.336 ± 0.005	$\mathbf{0.41 \pm 0.014}$	0.338 ± 0.007	$\mathbf{0.384 \pm 0.014}$
	Recall	0.357 ± 0.004	$\mathbf{0.459 \pm 0.016}$	0.359 ± 0.007	$\mathbf{0.426 \pm 0.016}$
	Accuracy	0.016 ± 0.003	$\mathbf{0.179 \pm 0.031}$	0.02 ± 0.005	$\mathbf{0.134 \pm 0.031}$

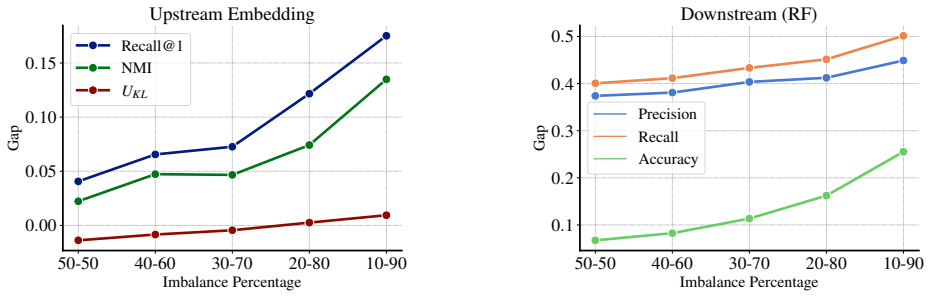


Figure 3: Impact of varying imbalance between the *minoritized* and *majoritized* classes on upstream embedding and downstream classifier (RF) in the manually class imbalanced CARS196 experiments. (Note: the imbalance percentage $50 - 50$ is equivalent to the balanced setting). Gaps increase both upstream and downstream with more imbalance introduced to the upstream training data.

than the control, even when manual imbalance is set at $40 - 60$ upstream, *and* the downstream training dataset is balanced. For results with additional downstream classifiers, see Supplemental. This experiment demonstrates that the propagation of bias to downstream will occur even with lower levels of imbalance, and does not appear to depend on the downstream classifier chosen.

6.3 Reduced subgroup gaps through partial de-correlation with sensitive attribute

Table 2a shows results for performance gaps between relevant subgroups in both facial recognition datasets. PARADE shows strong results for CUB200 bird color dataset, primarily reducing gaps downstream and accordingly to recall@1 (Definition 1). PARADE can reliably reduce gaps for both the representation space and downstream classifiers on LFW. Interestingly, we observe that the majoritized subgroup ("White") had worst performance of all "Race" subgroups (see Supplemental), contrary to previous results (Samadi et al., 2018).[3] As such, we measure gaps between the worst-performing subgroup and others.

[2]Due to the great number of singleton classes in LFW, recall@1 is discarded as a metric.

[3]Note: minoritized subgroups can still encounter notable bias across other axes more difficult to measure (Radford & Espenshade, 2014).

Table 2: *Comparison between PARADE and standard losses with distance-weighted sampling* of average gaps in representation space and downstream classification (logistic regressor) over 3 seeds between minoritized and majoritized groups in **(a)** facial dataset studies, namely on CelebA (w.r.t. "Fitzpatrick Skintone") and between worst-performing subgroup and other subgroups in LFW^2(w.r.t. "Race") with *Margin* loss and **(b)** bird color experiments for CUB200 image dataset (w.r.t. color) with *Margin* and *Triplet* loss. **Bold** represents smaller gap (better fairness performance).

(a)

Facial Datasets		**CelebA** (*skintone*)		**LFW** (*race*)	
		PARADE	Margin · Distance	PARADE	Margin · Distance
UPSTREAM EMBEDDING	Recall@1	$\mathbf{0.085 \pm 0.009}$	0.122 ± 0.005	0.075 ± 0.014	$\mathbf{0.068 \pm 0.013}$
	NMI	$\mathbf{-0.012 \pm 0.003}$	-0.002 ± 0.003	$\mathbf{0.041 \pm 0.003}$	0.048 ± 0.003
	U_{KL}	$\mathbf{-0.04 \pm 0.011}$	-0.03 ± 0.007	$\mathbf{0.163 \pm 0.003}$	0.165 ± 0.005
DOWNSTREAM CLASSIFICATION	Precision	0.146 ± 0.006	$\mathbf{0.1 \pm 0.007}$	$\mathbf{0.004 \pm 0.002}$	0.005 ± 0.005
	Recall	0.141 ± 0.007	$\mathbf{0.098 \pm 0.007}$	$\mathbf{0.003 \pm 0.001}$	0.007 ± 0.006
	Accuracy	0.131 ± 0.006	$\mathbf{0.082 \pm 0.005}$	$\mathbf{0.009 \pm 0.003}$	0.012 ± 0.009

(b)

CUB200-2011 *color*		PARADE (M · D)	Margin · Distance	PARADE (T · D)	Triplet · Distance
UPSTREAM EMBEDDING	Recall@1	$\mathbf{0.172 \pm 0.021}$	0.176 ± 0.041	$\mathbf{0.172 \pm 0.027}$	0.195 ± 0.051
	NMI	0.349 ± 0.031	$\mathbf{0.326 \pm 0.184}$	0.372 ± 0.291	$\mathbf{0.359 \pm 0.024}$
	U_{KL}	0.167 ± 0.013	$\mathbf{0.153 \pm 0.013}$	0.174 ± 0.035	$\mathbf{0.159 \pm 0.018}$
DOWNSTREAM CLASSIFICATION	Precision	$\mathbf{0.317 \pm 0.046}$	0.333 ± 0.049	$\mathbf{0.248 \pm 0.038}$	0.308 ± 0.119
	Recall	$\mathbf{0.352 \pm 0.039}$	0.363 ± 0.046	$\mathbf{0.276 \pm 0.042}$	0.337 ± 0.123
	Accuracy	0.163 ± 0.018	$\mathbf{0.153 \pm 0.028}$	$\mathbf{0.148 \pm 0.049}$	0.154 ± 0.029

For CelebA, we find for standard methods the minoritized subgroups to generally perform worst. PARADE excels at gap reduction upstream but encounters larger subgroup gaps downstream compared to standard methods. While PARADE does reduce downstream gaps between light skintones (I, II, and III), and the two lighter dark skintones (IV, V), gaps increase between lighter skintones and the darkest skintone (VI) (see Supplemental). Because skintone VI constitutes $< 1\%$ of the CelebA dataset, PARADE is likely not able to learn similarity between faces over attributes besides skintone. And PARADE is prevented from learning similarities based on skintone due to de-correlation. In such settings, PARADE could be combined with oversampling minoritized subgroups to ensure better performance.

In general, the results show promising benefits of PARADE to adequately address and improve on the challenge of subgroup gaps for DML models used in facial recognition; and in the standard DML dataset CUB200, for recall@1 upstream (Definition 1) and across metrics downstream (Table 2b).

7 DISCUSSION

In this work, we introduce the *finDML* benchmark, a framework for fairness in deep metric learning (§3.2). We demonstrate the fairness limitations of established DML techniques, and the surprising propagation of embedding space bias to downstream classifiers. Importantly, we find that this bias cannot be addressed at the level of downstream classifiers but instead needs to be addressed at the DML stage. We investigate the limit of this propagation in manually introduced imbalance, and finally show that PARADE can reduce subgroup gaps in several settings.

Limitations PARADE suffers from pitfalls similar to other "fairness with awareness" methods: PARADE uses information only on pre-defined sensitive attributes and therefore can be unfair w.r.t. other sensitive attributes. PARADE does have an advantage in addressing the combinatorial number of attributes considered in multi-attribute fairness through DML, which will scale sub-combinatorially in time/space complexity. We also note that subgroup gaps are not sufficient to capturing societal understandings of fairness, and there is no consensus as to how to remedy such gaps (Chouldechova & Roth, 2018; Dwork et al., 2012; Hardt et al., 2016; Zemel et al., 2013; Zafar et al., 2017). Additionally, while PARADE intentionally optimizes Definitions 1 and 2, we provide no explicit guarantee and optimization of uniformity, Definition 3, remains an open problem. Finally, PARADE does incur slight decrease in overall performance, similar to other methods (Wick et al., 2019) (see Supplemental for per-subgroup performance and additional fairness-utility trade-off analysis for PARADE).

Code of Ethics Statement The work presented here deals with fairness in deep metric learning. A portion of our studies in the paper focus on CARS196 and CUB200-2011 datasets, which have

consistently been used in benchmarking novel DML frameworks (Krause et al., 2013; Wah et al., 2011). The fairness analysis considered for CUB200-2011 deals with bird color, which does not, to our knowledge, correspond with any societal problems relating to fairness. Nonetheless, as CUB200-2011 is used in a litany of papers for SOTA performance comparison, *finDML* includes CUB200 so that DML methods can be analyzed w.r.t. fairness on a dataset used in their original paper.

We do include facial recognition datasets and tasks and analyze fairness with respect to facial attributes. Facial recognition does raise ethical concerns in practice. We note that our paper attempts to address primary social concerns in facial identity recognition. We do not encourage the task of facial *attribute* recognition, and solely use labeled attributes that correspond to known axes of bias for fairness analysis (e.g. Race and Skintone). As PARADE has solely been tested in two widely used public facial recognition datasets, we cannot guarantee fairness nor privacy in practical settings with private facial datasets.

Reproducibility Statement Additional experimental results discussed in the main paper and others are contained in Supplemental C. Implementation details including attribute information, generation of attributes, training parameters, metric calculation and gap computation are listed in Supplemental D. In our zipped supplemental materials file, we include all code used to generate the experiments. Individual scripts for generation of each experiment are contained in the "experiments" directory of the code file.

REFERENCES

Tameem Adel, Isabel Valera, Zoubin Ghahramani, and Adrian Weller. One-network adversarial fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2412–2420, 2019.

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. In *International Conference on Machine Learning*, pp. 159–168. PMLR, 2018.

Alexander Amini, Ava P Soleimany, Wilko Schwarting, Sangeeta N Bhatia, and Daniela Rus. Uncovering and mitigating algorithmic bias through learned latent structure. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 289–295, 2019.

Richard Berk. An impact assessment of machine learning risk forecasts on parole board decisions and recidivism. *Journal of Experimental Criminology*, 13(2):193–216, 2017.

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Souradip Chakraborty, Ekansh Verma, Saswata Sahoo, and Jyotishka Datta. Fairmixrep: Self-supervised robust representation learning for heterogeneous data with fairness constraints. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pp. 458–463. IEEE, 2020.

Irene Chen, Fredrik D Johansson, and David Sontag. Why is my classifier discriminatory? *arXiv preprint arXiv:1805.12002*, 2018a.

Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018b.

Victoria Cheng, Vinith M. Suriyakumar, Natalie Dullerud, Shalmali Joshi, and Marzyeh Ghassemi. Can you fake it until you make it? impacts of differentially private synthetic data on downstream classification fairness. In *Proceedings of the 2021 ACM Conference on Fairness,*

Accountability, and Transparency, FAccT '21, pp. 149–160, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445879. URL https://doi.org/10.1145/3442188.3445879.

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

Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4685–4694, 2019. doi: 10.1109/CVPR.2019.00482.

Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou. Deep adversarial metric learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2780–2789, 2018. doi: 10.1109/CVPR.2018.00294.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem Meent. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2525–2534. PMLR, 2019.

Xavier Gitiaux and Huzefa Rangwala. Fair representations by compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11506–11515, 2021.

Amir Globerson and Sam T. Roweis. Metric learning by collapsing classes. In Y. Weiss, B. Schölkopf, and J. C. Platt (eds.), *Advances in Neural Information Processing Systems 18*, pp. 451–458. MIT Press, 2006. URL http://papers.nips.cc/paper/2947-metric-learning-by-collapsing-classes.pdf.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19: 513–520, 2006.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning, 2016.

Ben Harwood, BG Kumar, Gustavo Carneiro, Ian Reid, Tom Drummond, et al. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2821–2829, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network, 2018.

J. Hu, J. Lu, and Y. Tan. Discriminative deep metric learning for face verification in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

Gary B. Huang, Vidit Jain, and Erik Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, 2007.

Christina Ilvento. Metric learning for individual fairness, 2020.

Pierre Jacob, David Picard, Aymeric Histace, and Edouard Klein. Metric learning with horde: High-order regularizer for deep embeddings. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Ayush Jaiswal, Rob Brekelmans, Daniel Moyer, Greg Ver Steeg, Wael AbdAlmageed, and Premkumar Natarajan. Discovery and separation of features for invariant representation learning. *arXiv preprint arXiv:1912.00646*, 2019.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2011.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2020.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018.

Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013.

Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pp. 365–372, 2009. doi: 10.1109/ICCV.2009.5459250.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H. Chi. Fairness without demographics through adversarially reweighted learning, 2020.

Hui Li, Maryellen L. Giger, Benjamin Q. Huynh, and Natalia O. Antropova. Deep learning in breast cancer risk assessment: evaluation of convolutional neural networks on a clinical dataset of full-field digital mammograms. *Journal of medical imaging (Bellingham, Wash.)*, 4(4):041304–041304, Oct 2017. ISSN 2329-4302. doi: 10.1117/1.JMI.4.4.041304. URL https://pubmed.ncbi.nlm.nih.gov/28924576.

Yujia Li, Kevin Swersky, and Richard Zemel. Learning unbiased features. *arXiv preprint arXiv:1412.5244*, 2014.

Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, pp. 714–729, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01267-0.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Trans. Information Theory*, 28:129–136, 1982.

Francesco Locatello, Gabriele Abbati, Tom Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. On the fairness of disentangled representations. *arXiv preprint arXiv:1905.13662*, 2019a.

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*, pp. 4114–4124. PMLR, 2019b.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

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

Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

Timo Milbich, Karsten Roth, Homanga Bharadhwaj, Samarth Sinha, Yoshua Bengio, Björn Ommer, and Joseph Paul Cohen. Diva: Diverse visual feature aggregation for deep metric learning. *CoRR*, abs/2004.13458, 2020. URL https://arxiv.org/abs/2004.13458.

Timo Milbich, Karsten Roth, Samarth Sinha, Ludwig Schmidt, Marzyeh Ghassemi, and Björn Ommer. Characterizing generalization under out-of-distribution shifts in deep metric learning. *CoRR*, abs/2107.09562, 2021. URL https://arxiv.org/abs/2107.09562.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Daniel Moyer, Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. Invariant representations without adversarial training. *arXiv preprint arXiv:1805.09458*, 2018.

Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim. A metric learning reality check. *CoRR*, abs/2003.08505, 2020. URL https://arxiv.org/abs/2003.08505.

William Paul and Philippe Burlina. Generalizing fairness: Discovery and mitigation of unknown sensitive attributes. *arXiv preprint arXiv:2107.13625*, 2021.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Alexandria Walton Radford and Thomas J Espenshade. *No longer separate, not yet equal: race and class in elite college admission and campus life*. Princeton University Press, 2014.

Borja Rodríguez-Gálvez, Ragnar Thobaben, and Mikael Skoglund. A variational approach to privacy and fairness. *arXiv preprint arXiv:2006.06332*, 2020.

Harrison Rosenberg, Brian Tang, Kassem Fawaz, and Somesh Jha. Fairness properties of face recognition and obfuscation systems, 2021.

Karsten Roth and Biagio Brattoli. Deep-metric-learning-baselines. `https://github.com/Confusezius/Deep-Metric-Learning-Baselines`, 2019.

Karsten Roth, Timo Milbich, and Bjorn Ommer. Pads: Policy-adapted sampling for visual similarity learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020a.

Karsten Roth, Timo Milbich, Björn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. S2sd: Simultaneous similarity-based self-distillation for deep metric learning, 2020b.

Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8242–8252. PMLR, 13–18 Jul 2020c. URL `http://proceedings.mlr.press/v119/roth20a.html`.

Samira Samadi, Uthaipon Tantipongpipat, Jamie Morgenstern, Mohit Singh, and Santosh Vempala. The price of fair pca: One extra dimension. *arXiv preprint arXiv:1811.00103*, 2018.

Mhd Hasan Sarhan, Nassir Navab, Abouzar Eslami, and Shadi Albarqouni. Fairness by learning orthogonal disentangled representations. In *European Conference on Computer Vision*, pp. 746–761. Springer, 2020.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Samarth Sinha, Karsten Roth, Anirudh Goyal, Marzyeh Ghassemi, Hugo Larochelle, and Animesh Garg. Uniform priors for data-efficient transfer, 2020.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pp. 1857–1865, 2016.

Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2164–2173. PMLR, 2019.

Yao-Hung Hubert Tsai, Martin Q Ma, Han Zhao, Kun Zhang, Louis-Philippe Morency, and Ruslan Salakhutdinov. Conditional contrastive learning: Removing undesirable information in self-supervised representations. *arXiv preprint arXiv:2106.02866*, 2021.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2593–2601, 2017.

Jiang Wang, Yang song, Thomas Leung, Chuck Rosenberg, Jinbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking, 2014.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2020.

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning, 2020a.

Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation, 2020b.

Kilian Q Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schölkopf, and J. C. Platt (eds.), *Advances in Neural Information Processing Systems 18*, pp. 1473–1480. MIT Press, 2006. URL http://papers.nips.cc/paper/2795-distance-metric-learning-for-large-margin-nearest-neighbor-classification.pdf.

Eric W. Weisstein. Hypersphere, 2002.

Michael Wick, swetasudha panda, and Jean-Baptiste Tristan. Unlocking fairness: a trade-off revisited. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/373e4c5d8edfa8b74fd4b6791d0cf6dc-Paper.pdf.

Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning, 2018.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. *arXiv preprint arXiv:1705.11122*, 2017.

Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 570–575. IEEE, 2018.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pp. 962–970. PMLR, 2017.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pp. 325–333. PMLR, 2013.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018.

Han Zhao and Geoffrey J Gordon. Inherent tradeoffs in learning fair representations. *arXiv preprint arXiv:1906.08386*, 2019.

Han Zhao, Amanda Coston, Tameem Adel, and Geoffrey J Gordon. Conditional learning of fair representations. *arXiv preprint arXiv:1910.07162*, 2019.

Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

SUPPLEMENTAL MATERIAL

A ADDITIONAL BACKGROUND

A.1 DEEP METRIC LEARNING DEFINITIONS

Here, we iterate through some common DML criteria and batch mining strategies more formally than in the main paper. Throughout this section, let X denote the input data, $\phi(X)$ the embedded data, Y the class label, and let $Y(x)$ denote the value of the ground truth class label for data instance x. Denote the set of all positive pairs with respect to class label Y as $\mathbb{P} = \{(x_1, x_2) \in X \times X : Y(x_1) = Y(x_2), x_1 \neq x_2\}$. Denote the set of all negative pairs with respect to class label Y as $\mathbb{N} = \{(x_1, x_2) \in X \times X : Y(x_1) \neq Y(x_2)\}$. We use the notation $(x_a, x_p, x_n) \in X \times X \times X$ to denote a triplet with an anchor sample x_a, positive sample x_p where $Y(x_a) = Y(x_p)$, and negative sample x_n where $Y(x_a) \neq Y(x_n)$.

Batch Sampling and Mining The batch sampling procedure in deep metric learning methods differ from that of generic deep classifiers in that canonical loss functions require tuples or pairs of samples in order to utilise ranking objectives as training surrogates to learn an appropriate similarity metric. To ensure that tuples with positive and negative examples can be extracted from the batch, the Samples-Per-Class-n (SPC-n) heuristic (see e.g. Roth et al. (2020c)) is generally used, where commonly $n = 2, 4, 8$. Given a batch size b, the SPC-n technique randomly selects b/n classes from which n training samples are then drawn randomly to be included in each batch $\mathcal{B}$.

After feeding the batch through the network, tuples are *mined* from the batch to use in the loss function. We refer to mining in this paper either as *batch mining* or overload the both as *batch sampling* terminology. The naive solution to tuple mining is random mining, in which all possible tuples of the form (x_a, x_p, x_n) are considered and b are randomly chosen from the batch. However, this method lacks the capacity to utilize valuable information about the current embedding space, and is prone to significant redundancy in the training signal Schroff et al. (2015); Wu et al. (2018).

Definition 4 (Random Mining)**.** *Hu et al. (2014) For each $x_a \in \mathcal{B}$, we randomly draw a positive example from $\{x_p \in \mathcal{B} : Y(x_p) = Y(x_a), x_p \neq x_a\}$ and a negative example from $\{x_n \in \mathcal{B} : Y(x_n) \neq Y(x_a)\}$ to form the triplet (x_a, x_p, x_n).*

Intuitively, this could be mitigated by *hard* mining heuristics searching for negative samples that are closer to the anchor sample in the embedding space than positive samples, thereby always ensuring a significant training signal. Unfortunately, such approaches are prone to heavy overfitting, training instability and large gradient variance, thereby commonly resulting in less-than-optimal solutions (see e.g. Schroff et al. (2015); Harwood et al. (2017); Wu et al. (2018)). Recent approaches thus establish more lenient heuristics, such as through the introduction of slack parameters to the hard mining objective (e.g. semi-hard mining Schroff et al. (2015) or softhard mining Roth & Brattoli (2019)).

Definition 5 (Semi-hard Mining)**.** *For each $x_a \in \mathcal{B}$, we randomly draw a positive example from $\{x_p \in \mathcal{B} : Y(x_p) = Y(x_a), x_p \neq x_a\}$, and a negative example from the set*

$$\{x_n \in \mathcal{B} : Y(x_n) \neq Y(x_a), \|\phi(x_a) - \phi(x_n)\|_2^2 \|\phi(x_a) - \phi(x_p) + \gamma\|_2^2\}$$

where $\gamma \in \mathbb{R}$ is a slack parameter, to form the triplet (x_a, x_p, x_n).

While other adaptive means (e.g. Harwood et al. (2017); Roth et al. (2020a)) have shown strong performance improvements, modern predefined heuristics such as distance-weighted tuple mining Wu et al. (2018) offer a better cost-to-performance tradeoff Roth et al. (2020a). Here, the heuristic leverages the fact that embeddings are commonly normalized to have unit L_2 norm for regularization purposes Wu et al. (2018). This ensures a distribution over a unit hypersphere, in which explicit pairwise distributions can be established Weisstein (2002); Wu et al. (2018). By inverting this distribution, distance-weighted mining can thus encourage a much more diverse coverage of tuple difficulties, improving generalization performance and reducing gradient variance Wu et al. (2018).

Definition 6 (Distance-weighted)**.** *For embedding spaces normalized to the $(D-1)$-dimensional hypersphere $\mathcal{S}^{D-1}$, we haveWeisstein (2002); Wu et al. (2018) the following pairwise sampling distribution $q(\bullet,\bullet)$:*

$$q\left(d\left(\phi(x_i),\phi(x_j)\right)\right) \propto d\left(\phi(x_i),\phi(x_j)\right)^{D-2}\left[1-\frac{1}{4}d\left(\phi(x_i),\phi(x_j)\right)\right]^{\frac{D-3}{2}}$$

for embedding pairs $(\phi(x_i),\phi(x_j)) \in \mathcal{S}^{D-1}$ and Euclidean distance $d(\bullet,\bullet)$. For each $x_a \in \mathcal{B}$, we randomly draw a positive example from $\{x_p \in \mathcal{B} : Y(x_p) = Y(x_a), x_p \neq x_a\}$, and sample a negative example based on an inverse distance distribution w.r.t. q:

$$P(x_n|x_a) \propto \textit{min}(\lambda, q^{-1}(d(\phi(x_i),\phi(x_j))))$$

where $\lambda \in \mathbb{R}$ defines a clipping parameter to avoid potentially erroneous training samples.

Examined Objectives The primary goal of DML loss functions is to provide a training surrogate that implicitly optimizes for desired metric space quantities by narrowing down the expected distance between positive pairs of samples and expanding on the expected distance between negative pairs of samples in the embedding space. Most commonly employed pair Hadsell et al. (2006) and tripled-based Schroff et al. (2015); Hoffer & Ailon (2018) ranking losses penalize close negative pairs and disparate positive pairs up to a predefined margin to avoid overclustering. Using $\mathbb{P}(x)$ to denote all positive pairs containing x

$$\mathbb{P}(x) = \{(x_1,x_2) \in \mathbb{P} : x_1 = x\}$$

and $\mathbb{N}(x)$ to denote all negative pairs containing x

$$\mathbb{N}(x) = \{(x_1,x_2) \in \mathbb{N} : x_1 = x\}$$

we define

Definition 7 (Contrastive)**.** *Hadsell et al. (2006) Given a batch $\mathcal{B}$, and pairs of samples $\mathbb{S}$ over $\mathcal{B}\times\mathcal{B}$, the contrastive objective is defined as:*

$$\mathcal{L}_{\textit{contr}} = \frac{1}{b}\sum_{(x_i,x_j)\in\mathbb{S}} \mathbb{I}_{Y(x_i)=Y(x_j)} d\left(\phi(x_i),\phi(x_j)\right) + \mathbb{I}_{Y(x_i)\neq Y(x_j)}\left[\gamma - d\left(\phi(x_i),\phi(x_j)\right)\right]_+$$

with margin γ.

Definition 8 (Triplet)**.** *Hoffer & Ailon (2018) The triplet loss extends the contrastive objective with sample triplets and can be defined as:*

$$\mathcal{L}_{\textit{tripl}} = \frac{1}{b}\sum_{\substack{(x_a,x_p,x_n)\in\mathcal{T}\\ Y(x_a)=Y(x_p)\neq Y(x_n)}} \left[d\left(\phi(x_a),\phi(x_p)\right) - d\left(\phi(x_a),\phi(x_n)\right) + \gamma\right]_+$$

with margin γ.

Margin loss extends the triplet objective through the inclusion of a learnable boundary β between positive and negative pairs Wu et al. (2018). In our experiments, we utilise $\beta = 1.2$. These criteria are widely used (see e.g. Roth et al. (2020c); Musgrave et al. (2020)) and require mining to make use of the batch information.

Definition 9 (Margin)**.** *Wu et al. (2018) The margin objective integrates the learnable distance boundary β between positive and negative pairs of samples for a relative ordering of pairs with respect to β as*

$$\mathcal{L}_{\textit{margin}} = \sum_{(x_i,x_j)\in\mathbb{S}} \gamma + \mathbb{I}_{Y(x_i)=Y(x_j)}\left(d\left(\phi(x_i),\phi(x_j)\right) - \beta\right) - \mathbb{I}_{Y(x_i)\neq Y(x_j)}\left(d\left(\phi(x_i),\phi(x_j)\right) - \beta\right)$$

Going beyond pairs and triplets, one can also consider the case of more general n-tuples, which was investigated e.g. in the N-Pair objective Sohn (2016) and the Multisimilarity loss Wang et al. (2020a).

Definition 10 (N-Pair)**.** *Sohn (2016) N-Pair loss is a simple augmentation of the triplet framework in which all negatives in the batch $\mathcal{B}$ are incorporated in the objective function as:*

$$\mathcal{L}_{npair} = \frac{1}{b} \sum_{\substack{(x_a,x_p)\in\mathcal{B} \\ Y(x_a)=Y(x_p), a\neq p}} \log\left(1 + \sum_{\substack{x_n\in\mathcal{B} \\ Y(x_a)\neq Y(x_n)}} \exp\left(\phi(x_a)^{*,T}\phi(x_n) - \phi(x_a)^{*,T}\phi(x_p)^*\right)\right) + \frac{\nu}{b}\cdot\sum_{i\in\mathcal{B}}\|\phi(x_i)^*\|_2^2 \quad (6)$$

where ν denotes an embedding regularization parameter due to slow convergence for normalized embeddings stated in Sohn (2016)

Definition 11 (Multisimilarity)**.** *Wang et al. (2020a) Multisimilarity loss fits into the ranking loss category, but in addition to evaluation of cosine similarity between positive-anchor pairs and negative-anchor pairs, the objective evaluates positive-positive and negative-negative pairs with respect to the anchor:*

$$s_c^*(x_i,x_j) = \begin{cases} s_c\left(\phi(x_i),\phi(x_j)\right) & s_c\left(\phi(x_i),\phi(x_j)\right) > \min_{x_j\in\mathbb{P}(x_i)} s_c\left(\phi(x_i),\phi(x_j)\right) - \epsilon \\ s_c\left(\phi(x_i),\phi(x_j)\right) & s_c\left(\phi(x_i),\phi(x_j)\right) < \max_{x_k\in\mathbb{N}(x_i)} s_c\left(\phi(x_i),\phi(x_k)\right) + \epsilon \\ 0 & \textit{otherwise} \end{cases}$$

$$\mathcal{L}_{multisim} = \frac{1}{b}\sum_{x_i\in\mathcal{B}}\frac{1}{\alpha}\log\left[1 + \sum_{x_j\in\mathbb{P}(x_i)}\exp\left(-\alpha\left(s_c^*\left(\phi(x_i),\phi(x_j)\right) - \lambda\right)\right)\right] + \frac{1}{\beta}\log\left[1 + \sum_{k\in\mathbb{N}(x_i)}\exp\left(\beta\left(s_c^*\left(\phi(x_i),\phi(x_k)\right) - \lambda\right)\right)\right] \quad (7)$$

where cosine similarity $s_c(x,y) = x^T y$ for two normalized vectors $x, y \in X$.

Notably, the Multisimilarity loss employs a masking process as a stand-in for the lack of batch-mining heuristic. While this proves to be similarly successfull in addressing the tuple sampling complexity issue, this can also be addressed through the usage of proxy-samples. These are dummy variables that represent various contextual properties (such as mean class representations) to serve as standing for actual samples, which is found e.g. in the ArcFace Deng et al. (2019) or ProxyNCA loss Movshovitz-Attias et al. (2017).

Definition 12 (Proxy-NCA)**.** *Kim et al. (2020) ProxyNCA learns class proxies, or class centers, which each represent a class in the set of unique classes $\mathcal{Y}$. Then, each anchor from the batch is sampled and a positive or negative proxy $\psi^c \in \mathbb{R}^d$ per class $c \in \mathcal{Y}$ is introduced in lieu of a positive or negative sample, respectively, giving:*

$$\mathcal{L}_{proxy} = -\frac{1}{b}\sum_{x_i\in\mathcal{B}}\log\left(\frac{\exp\left(-d\left(\phi(x_i),\psi_{Y(x_i)}\right)\right)}{\sum_{c\in\mathcal{Y}\setminus\{Y(x_i)\}}\exp\left(-d\left(\phi(x_i),\psi_c\right)\right)}\right)$$

Definition 13 (Arcface)**.** *Deng et al. (2019) Arcface combines proxy and angular loss methods (e.g. in Wang et al. (2017)) to enforce an angular margin between the embeddings ϕ and a proxy (or approximate center) $W \in \mathbb{R}^{c\times d}$ for each class, giving the following:*

$$\mathcal{L}_{arc} = -\frac{1}{b}\sum_{x_i\in\mathcal{B}}\log\frac{\exp\left(s\cdot\cos\left(W_{Y(x_i)}^T\phi(x_i) + \gamma = 0.5\right)\right)}{\exp\left(s\cdot\cos\left(W_{Y(x_i)}^T\phi(x_i) + \gamma = 0.5\right)\right) + \sum_{\substack{x_j\in\mathcal{B} \\ Y(x_i)\neq Y(x_j)}}\exp\left(s\cdot\cos\left(W_{Y(x_j)}^T\phi(x_i)\right)\right)}$$

where the angular component is encoded in additive angular margin penalty γ, and s is a scaling parameter, which denotes the radius of the effective utilized hypersphere $\mathcal{S}$.

Standard Performance Metrics Performance metrics in deep metric learning aim to capture the quality of the similarity metric learned by the deep embedding model. Therefore, standard performance metrics in DML reflect the closeness between samples of the same class, the separability

of samples of different classes, the clustering quality of embedding, and the uniformity over the hypersphere embedding space, which has been linked to zero-shot generalization capability Wang & Isola (2020), as discussed in Section 2. In our experiments, we utilize recall@1 Jegou et al. (2011), normalized mutual information Manning et al. (2010) between cluster labels assigned by the well-known K-Means Lloyd (1982) algorithm and ground-truth class labels, and U_{KL} to measure the closeness between samples of same class, cluster quality of the embedding (and hence, the separability of distinct classes) and uniformity, respectively. Here, we define these metrics formally, but we note that there exist multitudinous performance metrics for DML that we do not define here or use explicitly for our results, including f1 score, mean average precision (mAP), and recall@k for $k > 1$ Jegou et al. (2011).

Definition 14 (Recall@k). *Jegou et al. (2011) Given $k \in \{1, \ldots, |X|\}$, denote NN_k as defined in Definition 1. Then, Recall@k is measured as:*

$$\textit{Recall@k} = \frac{1}{|X|} \sum_{x \in X} \begin{cases} 1 & \exists \tilde{x} \in NN_k(x) : Y(\tilde{x}) = Y(x) \\ 0 & \textit{else} \end{cases}$$

Definition 15 (Normalized Mutual Information Score on Clusters). *Manning et al. (2010) Let C a clustering algorithm, such as K-Means Lloyd (1982) with the number of clusters set to $|Y|$, such that $C(x)$ indicates the cluster label for data point $x \in X$. The normalized mutual information score between the target labels Y and the cluster labels C is measured as:*

$$NMI = \frac{2 \cdot I(Y(X); C(X))}{H(Y(X)) + H(C(X))}$$

where for random variables X, Y, $I(\cdot, \cdot)$ denotes the mutual information function:

$$I(X; Y) = H(Y) - H(Y|X)$$

and $H(\cdot)$ denotes the entropy function:

$$H(X) = -\sum_{x \in X} \Pr(x) \log(\Pr(x))$$

The performance metric U_{KL}, used to measure feature uniformity for our empirical evaluations, is defined in Section 3.1.

A.2 Classification Fairness Definitions

Fairness definitions and criteria in classification are briefly mentioned in Section 2 of the main paper. Here, we provide explicit formulas for the most common fairness definitions, including demographic parity, equalized odds, and equality of opportunity Hardt et al. (2016), and provide some additional context on fairness definition evolution.

Definition 16 (Demographic Parity). *The predictor $\hat{Y}$ satisfies demographic parity with respect to attribute A and class Y if the predictor is independent of A:*

$$\Pr[\hat{Y} = 1 | A = a] = \Pr[\hat{Y} = 1 | A = b] \qquad \forall a, b \in A$$

Specifically, demographic parity has largely been used over the years as a simple and intuitive definition of fairness, in which a classifier is said to satisfy demographic parity if the sensitive attribute is independent of the output of the classifier. While demographic parity provides a simple fairness definition, the measure cannot capture fairness in classification tasks where the ground-truth label is inherently related to a certain attribute value Li et al. (2017).

Definition 17 (Equalized Odds). *The predictor $\hat{Y}$ satisfies demographic parity with respect to attribute A and class Y if the predictor is independent of A conditional on Y:*

$$\Pr[\hat{Y} = 1 | A = a, Y = y] = \Pr[\hat{Y} = 1 | A = b, Y = y] \qquad \forall a, b \in A, \forall y \in \{0, 1\}$$

from Hardt et al. (2016).

Definition 18 (Equality of Opportunity). *The predictor $\hat{Y}$ satisfies demographic parity with respect to attribute A and class Y if the predictor is independent of A conditional on positively labelled Y:*

$$\Pr[\hat{Y} = 1 | A = a, Y = 1] = \Pr[\hat{Y} = 1 | A = b, Y = 1] \qquad \forall a, b \in A$$

from Hardt et al. (2016).

This lead to the introduction of other fairness definitions that capture such nuances, the most well-known of which are probably equalized odds and equality of opportunity Hardt et al. (2016). However, fairness metrics overall have been criticized due to the choice of protected attribute over which to measure, and the inability of these metrics to capture bias with respect to certain attributes which are not known at test-time. We discuss this to a limited extent in Section 7.

B Dataset Summary Statistics

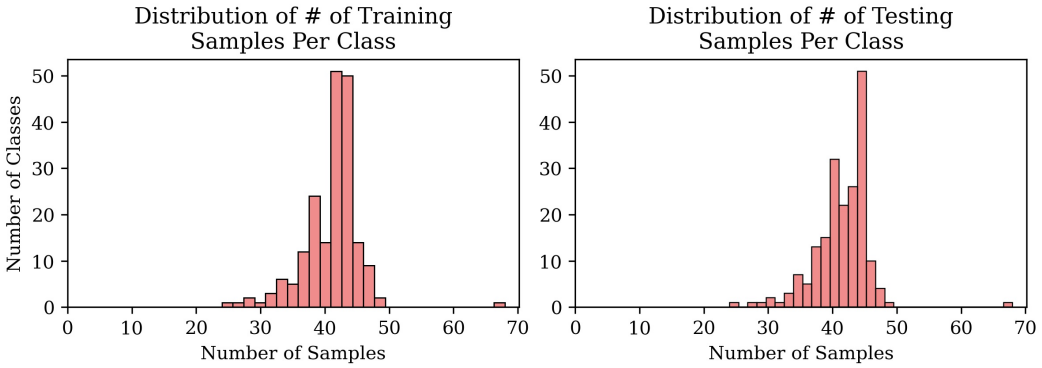


Figure 4: *Class distribution in CARS196.* Histograms visualizing the distribution over number of samples per class in the train (left) and test (right) datasets in CARS196.

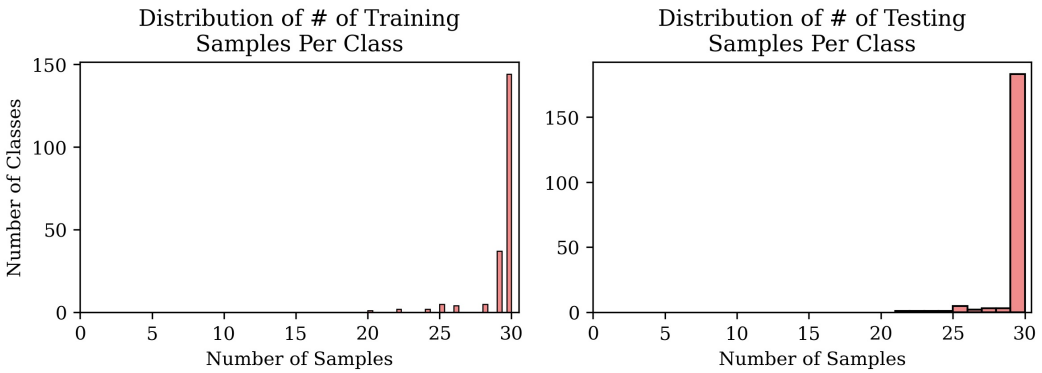


Figure 5: *Class distribution in CUB200.* Histograms visualizing the distribution over number of samples per class in the train (left) and test (right) datasets in CUB200.

	Black	Blue	Brown	Buff	Green	Grey	Iridescent	Olive	Orange	Red	White	Yellow
Train	21.20	5.58	18.08	3.01	0.37	19.20	0.51	0.49	1.02	3.52	13.35	13.65
Test	21.17	5.56	18.11	3.04	0.39	19.21	0.51	0.51	1.01	3.52	13.29	13.68

Table 3: *Summary statistics for CUB200 bird color* The percentage of the dataset constituted by each bird color in CUB200, in the train dataset and test dataset, respectively.

C Additional Results

C.1 CARS196

Additional results for all loss and batch mining strategies for the manually class imbalanced experiments and balanced controls for CARS196 are located in Tables 6 and 7. K-Means was also

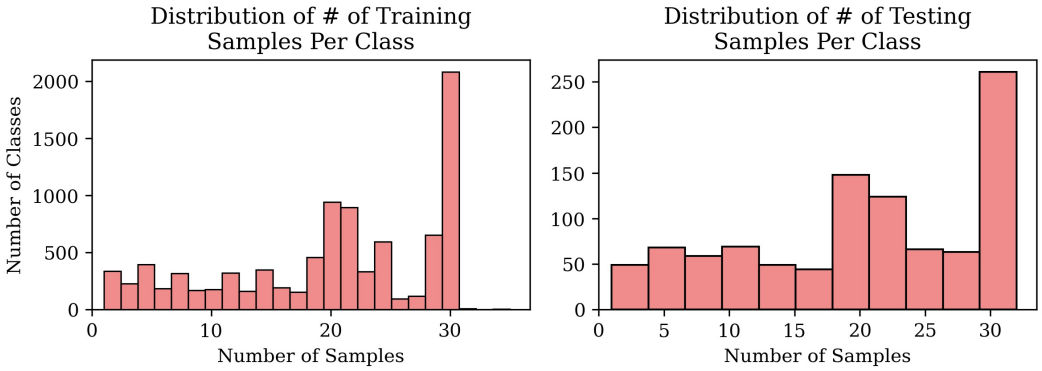


Figure 6: *Class distribution in CelebA.* Histograms visualizing the distribution over number of samples per class in the train (left) and test (right) datasets in CelebA.

	I	II	III	IV	V	VI
Train	1.10	32.04	47.92	15.12	3.20	0.61
Test	1.24	32.09	48.09	14.81	3.22	0.55

Table 4: *Summary statistics for CelebA Fitzpatrick Skintone.* The percentage of the dataset constituted by each Fitzpatrick Skintone in CelebA, in the train dataset and test dataset, respectively.

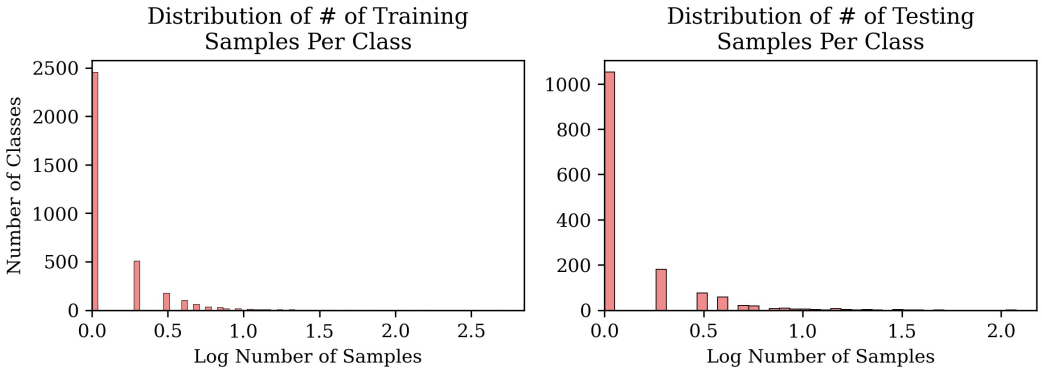


Figure 7: *Class distribution in LFW.* Histograms visualizing the distribution over logarithm of number of samples per class in the train (left) and test (right) datasets in LFW.

	Asian	Black	Indian	White
Train	8.43	4.17	1.71	85.70
Test	6.27	4.61	1.79	87.33

Table 5: *Summary statistics for LFW Race* The percentage of the dataset constituted by each Race in LFW, in the train dataset and test dataset, respectively.

tested as a downstream classifier but showed poor performance. The impact of varying imbalance in the manually class imbalanced CARS196 experiments with all tested downstream classifiers is displayed in Table 8. Additional results for benchmarking of further fairness improvement methods in downstream classification of "imbalanced" embeddings (aside from naive use of balanced datasets) are shown in Table 8.

			Overall					
			Contrastive · Distance		Margin · Distance		Margin · Semi-hard	
			Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced
UPSTREAM EMBEDDING		Recall@1	0.861 ± 0.003	0.83 ± 0.005	0.854 ± 0.002	0.819 ± 0.008	0.83 ± 0.002	0.811 ± 0.006
		NMI	0.909 ± 0.003	0.879 ± 0.003	0.894 ± 0.003	0.867 ± 0.005	0.876 ± 0.004	0.861 ± 0.007
		U_{KL}	0.433 ± 0.004	0.457 ± 0.008	0.096 ± 0.002	0.091 ± 0.002	0.133 ± 0.004	0.133 ± 0.003
DOWNSTREAM CLASSIFICATION	LR	Accuracy	0.878 ± 0.002	0.848 ± 0.006	0.88 ± 0.002	0.861 ± 0.004	0.858 ± 0.005	0.853 ± 0.005
		Precision	0.877 ± 0.002	0.848 ± 0.007	0.883 ± 0.002	0.864 ± 0.004	0.86 ± 0.005	0.856 ± 0.005
		Recall	0.876 ± 0.002	0.846 ± 0.006	0.879 ± 0.002	0.86 ± 0.004	0.856 ± 0.005	0.852 ± 0.005
	RF	Accuracy	0.855 ± 0.002	0.832 ± 0.006	0.816 ± 0.003	0.758 ± 0.011	0.819 ± 0.005	0.79 ± 0.006
		Precision	0.859 ± 0.003	0.835 ± 0.006	0.82 ± 0.003	0.763 ± 0.01	0.822 ± 0.005	0.794 ± 0.006
		Recall	0.855 ± 0.003	0.831 ± 0.006	0.815 ± 0.003	0.757 ± 0.011	0.817 ± 0.005	0.788 ± 0.005
	SVM	Accuracy	0.876 ± 0.002	0.852 ± 0.006	0.882 ± 0.002	0.863 ± 0.004	0.863 ± 0.005	0.86 ± 0.004
		Precision	0.875 ± 0.002	0.855 ± 0.007	0.888 ± 0.002	0.875 ± 0.003	0.867 ± 0.005	0.867 ± 0.004
		Recall	0.874 ± 0.002	0.85 ± 0.006	0.881 ± 0.002	0.863 ± 0.004	0.863 ± 0.005	0.86 ± 0.004

			Overall							
			Multisimilarity		Proxy-NCA		Triplet · Distance		Triplet · Semi-hard	
			Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced
UPSTREAM EMBEDDING		Recall@1	0.858 ± 0.002	0.839 ± 0.005	0.887 ± 0.001	0.858 ± 0.005	0.866 ± 0.003	0.848 ± 0.006	0.794 ± 0.003	0.778 ± 0.006
		NMI	0.898 ± 0.003	0.881 ± 0.004	0.915 ± 0.003	0.89 ± 0.005	0.903 ± 0.001	0.884 ± 0.005	0.849 ± 0.001	0.834 ± 0.007
		U_{KL}	0.151 ± 0.002	0.155 ± 0.003	0.083 ± 0.001	0.094 ± 0.003	0.304 ± 0.003	0.293 ± 0.003	0.397 ± 0.009	0.382 ± 0.007
DOWNSTREAM CLASSIFICATION	LR	Accuracy	0.886 ± 0.001	0.875 ± 0.005	0.898 ± 0.003	0.877 ± 0.005	0.885 ± 0.002	0.872 ± 0.004	0.828 ± 0.004	0.818 ± 0.005
		Precision	0.889 ± 0.001	0.878 ± 0.004	0.901 ± 0.002	0.879 ± 0.005	0.888 ± 0.002	0.874 ± 0.003	0.829 ± 0.003	0.821 ± 0.005
		Recall	0.885 ± 0.001	0.873 ± 0.005	0.898 ± 0.003	0.876 ± 0.005	0.883 ± 0.002	0.87 ± 0.004	0.825 ± 0.003	0.816 ± 0.005
	RF	Accuracy	0.825 ± 0.004	0.794 ± 0.005	0.852 ± 0.003	0.823 ± 0.007	0.858 ± 0.003	0.836 ± 0.004	0.808 ± 0.004	0.792 ± 0.008
		Precision	0.831 ± 0.004	0.797 ± 0.006	0.857 ± 0.003	0.825 ± 0.008	0.861 ± 0.003	0.838 ± 0.004	0.811 ± 0.004	0.795 ± 0.008
		Recall	0.824 ± 0.004	0.793 ± 0.005	0.852 ± 0.003	0.823 ± 0.008	0.857 ± 0.003	0.835 ± 0.004	0.807 ± 0.004	0.791 ± 0.008
	SVM	Accuracy	0.888 ± 0.001	0.876 ± 0.004	0.894 ± 0.002	0.871 ± 0.003	0.887 ± 0.002	0.878 ± 0.004	0.835 ± 0.003	0.83 ± 0.005
		Precision	0.893 ± 0.001	0.886 ± 0.004	0.902 ± 0.001	0.887 ± 0.003	0.892 ± 0.002	0.884 ± 0.004	0.839 ± 0.003	0.834 ± 0.005
		Recall	0.887 ± 0.001	0.876 ± 0.004	0.894 ± 0.002	0.871 ± 0.003	0.886 ± 0.003	0.877 ± 0.003	0.834 ± 0.003	0.829 ± 0.005

Table 6: *Overall results on CARS196.* Metrics over entire test dataset in representation space and downstream classification (LR, RF, and SVM) over 10 seed in manually class imbalanced experiments (Imbalanced) and control experiments (Balanced) for CARS196.

			Subgroup Gap					
			Contrastive · Distance		Margin · Distance		Margin · Semi-hard	
			Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced
UPSTREAM EMBEDDING		Recall@1	0.861 ± 0.003	0.83 ± 0.005	0.854 ± 0.002	0.819 ± 0.008	0.83 ± 0.002	
		NMI	-0.013 ± 0.004	0.106 ± 0.013	-0.016 ± 0.005	0.124 ± 0.014	-0.018 ± 0.006	0.11 ± 0.016
		U_{KL}	-0.093 ± 0.004	0.011 ± 0.011	-0.033 ± 0.002	0.01 ± 0.003	-0.038 ± 0.006	0.012 ± 0.005
DOWNSTREAM CLASSIFICATION	LR	Accuracy	0.002 ± 0.003	0.147 ± 0.023	0.003 ± 0.003	0.12 ± 0.013	0.004 ± 0.007	0.115 ± 0.018
		Precision	0.336 ± 0.004	0.407 ± 0.013	0.355 ± 0.008	0.402 ± 0.013	0.353 ± 0.008	0.403 ± 0.013
		Recall	0.351 ± 0.004	0.439 ± 0.016	0.368 ± 0.008	0.426 ± 0.015	0.367 ± 0.008	0.431 ± 0.014
	RF	Accuracy	0.001 ± 0.006	0.115 ± 0.021	0.002 ± 0.004	0.315 ± 0.022	0.002 ± 0.008	0.231 ± 0.024
		Precision	0.358 ± 0.005	0.396 ± 0.013	0.374 ± 0.006	0.441 ± 0.013	0.362 ± 0.007	0.429 ± 0.014
		Recall	0.373 ± 0.005	0.409 ± 0.015	0.387 ± 0.006	0.502 ± 0.013	0.376 ± 0.008	0.481 ± 0.016
	SVM	Accuracy	0.003 ± 0.004	0.086 ± 0.022	0.002 ± 0.003	0.039 ± 0.013	0.002 ± 0.006	0.055 ± 0.018
		Precision	0.33 ± 0.006	0.332 ± 0.023	0.35 ± 0.008	0.283 ± 0.024	0.347 ± 0.011	0.328 ± 0.018
		Recall	0.347 ± 0.004	0.338 ± 0.027	0.363 ± 0.008	0.27 ± 0.027	0.361 ± 0.011	0.327 ± 0.018

			Subgroup Gap							
			Multisimilarity		Proxy-NCA		Triplet · Distance		Triplet · Semi-hard	
			Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced
UPSTREAM EMBEDDING		Recall@1	0.858 ± 0.002	0.839 ± 0.005	0.887 ± 0.001	0.858 ± 0.005	0.866 ± 0.003	0.848 ± 0.006	0.794 ± 0.003	0.778 ± 0.006
		NMI	-0.014 ± 0.004	0.116 ± 0.014	-0.011 ± 0.004	0.148 ± 0.014	-0.013 ± 0.001	0.109 ± 0.016	-0.018 ± 0.002	0.083 ± 0.015
		U_{KL}	-0.03 ± 0.003	0.005 ± 0.004	-0.129 ± 0.002	0.005 ± 0.005	-0.047 ± 0.005	0.011 ± 0.006	-0.034 ± 0.013	0.023 ± 0.011
DOWNSTREAM CLASSIFICATION	LR	Accuracy	0.002 ± 0.002	0.117 ± 0.018	0.003 ± 0.005	0.155 ± 0.019	0.001 ± 0.004	0.131 ± 0.017	0.004 ± 0.005	0.12 ± 0.016
		Precision	0.352 ± 0.006	0.396 ± 0.014	0.334 ± 0.006	0.438 ± 0.012	0.343 ± 0.005	0.404 ± 0.011	0.357 ± 0.004	0.392 ± 0.009
		Recall	0.365 ± 0.006	0.418 ± 0.018	0.347 ± 0.006	0.46 ± 0.016	0.356 ± 0.005	0.431 ± 0.013	0.371 ± 0.004	0.421 ± 0.01
	RF	Accuracy	0.0 ± 0.006	0.26 ± 0.023	0.0 ± 0.005	0.221 ± 0.022	0.0 ± 0.004	0.168 ± 0.015	0.003 ± 0.005	0.141 ± 0.02
		Precision	0.378 ± 0.005	0.441 ± 0.011	0.379 ± 0.005	0.458 ± 0.01	0.361 ± 0.006	0.425 ± 0.008	0.361 ± 0.004	0.393 ± 0.012
		Recall	0.389 ± 0.005	0.485 ± 0.009	0.39 ± 0.005	0.476 ± 0.011	0.375 ± 0.006	0.45 ± 0.009	0.374 ± 0.004	0.426 ± 0.012
	SVM	Accuracy	0.002 ± 0.002	0.047 ± 0.015	0.0 ± 0.003	0.076 ± 0.013	0.0 ± 0.004	0.063 ± 0.018	0.002 ± 0.005	0.073 ± 0.02
		Precision	0.349 ± 0.006	0.267 ± 0.03	0.335 ± 0.006	0.29 ± 0.026	0.339 ± 0.007	0.297 ± 0.03	0.354 ± 0.007	0.364 ± 0.015
		Recall	0.362 ± 0.005	0.258 ± 0.032	0.349 ± 0.006	0.273 ± 0.029	0.353 ± 0.007	0.295 ± 0.03	0.368 ± 0.006	0.372 ± 0.016

Table 7: *Gap study on CARS196.* Average gaps in representation space and downstream classification (LR, RF, and SVM) over 10 seeds between minoritized and majoritized classes in manually class imbalanced experiments (Imbalanced) and control experiments (Balanced) for CARS196.

C.2 CUB200

Additional results for all loss and batch mining strategies for the manually class imbalanced experiments and balanced controls for CUB200 are located in Tables 9 and 10. K-Means was also tested as a downstream classifier but showed poor performance. The impact of varying imbalance in the manually class imbalanced experiments in the upstream embedding, and all tested downstream classifiers is displayed in Table 9. Additional results for benchmarking of further fairness improvement methods

Table 8: *Benchmarking additional fairness improvement methods in downstream classification on CARS196 (Classes).* Overall performance and subgroup gaps for Domain-Independent Training and Oversampling (Wang et al., 2020b) on CARS196 in class imbalanced experiments with upstream embedding trained on imbalanced dataset.

(a) Domain-Independent Training

METRIC ↓	Contr. (D)	Margin (D)	Margin (Sem.)	Msim.	ProxyNCA	Triplet (D)	Triplet (S)
Overall							
ACCURACY	0.812 ± 0.011	0.834 ± 0.009	0.820 ± 0.008	0.842 ± 0.008	0.869 ± 0.005	0.836 ± 0.007	0.742 ± 0.007
PRECISION	0.834 ± 0.009	0.861 ± 0.004	0.847 ± 0.004	0.872 ± 0.004	0.878 ± 0.005	0.865 ± 0.009	0.804 ± 0.008
RECALL	0.811 ± 0.011	0.833 ± 0.009	0.818 ± 0.008	0.840 ± 0.008	0.869 ± 0.005	0.834 ± 0.008	0.740 ± 0.008
Gap							
ACCURACY	0.001 ± 0.027	0.010 ± 0.018	0.017 ± 0.021	0.018 ± 0.018	0.120 ± 0.018	0.022 ± 0.017	0.094 ± 0.021
PRECISION	0.304 ± 0.021	0.275 ± 0.021	0.313 ± 0.019	0.247 ± 0.027	0.398 ± 0.015	0.236 ± 0.022	0.289 ± 0.016
RECALL	0.260 ± 0.024	0.218 ± 0.022	0.258 ± 0.018	0.182 ± 0.027	0.398 ± 0.018	0.175 ± 0.022	0.177 ± 0.016

(b) Oversampling

METRIC ↓	Contr. (D)	Margin (D)	Margin (Sem.)	Msim.	ProxyNCA	Triplet (D)	Triplet (S)
Overall							
ACCURACY	0.851 ± 0.007	0.862 ± 0.004	0.853 ± 0.006	0.875 ± 0.004	0.878 ± 0.004	0.875 ± 0.005	0.820 ± 0.005
PRECISION	0.854 ± 0.006	0.864 ± 0.004	0.855 ± 0.006	0.877 ± 0.004	0.880 ± 0.005	0.876 ± 0.004	0.822 ± 0.005
RECALL	0.853 ± 0.006	0.862 ± 0.004	0.853 ± 0.006	0.875 ± 0.004	0.878 ± 0.004	0.875 ± 0.004	0.821 ± 0.005
Gap							
ACCURACY	0.128 ± 0.023	0.099 ± 0.014	0.102 ± 0.020	0.097 ± 0.019	0.136 ± 0.017	0.108 ± 0.018	0.109 ± 0.019
PRECISION	0.398 ± 0.012	0.386 ± 0.017	0.391 ± 0.014	0.383 ± 0.015	0.422 ± 0.014	0.387 ± 0.013	0.387 ± 0.011
RECALL	0.423 ± 0.015	0.403 ± 0.017	0.414 ± 0.014	0.396 ± 0.019	0.436 ± 0.017	0.406 ± 0.015	0.413 ± 0.012

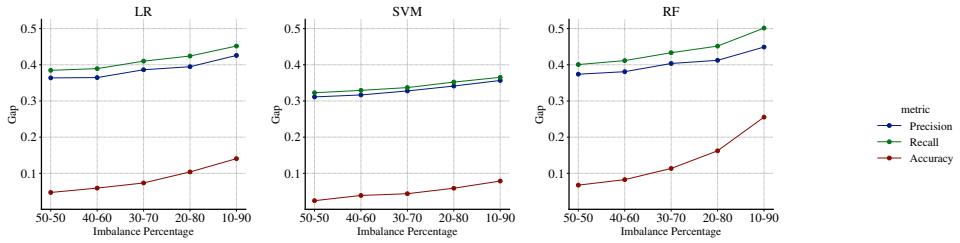


Figure 8: Impact of varying imbalance between the *minoritized* and *majoritized* classes on various downstream classifiers (RF, LR and SVM) in the manually class imbalanced CARS196 experiments. (Note: the imbalance percentage $50 - 50$ is equivalent to the balanced setting). Gaps increase for all classifiers downstream with more imbalance introduced to the upstream training data.

in downstream classification of "imbalanced" embeddings (aside from naive use of balanced datasets) are shown in Table 11. Benchmarking of fairness improvement methods in downstream classification for bird color are shown in Table 12. Per-subgroup and overall results for CUB200 color experiments with standard margin-distance and PARADE are displayed in Table 13.

C.3 CELEBA

Additional results for all loss and batch mining strategies for the CelebA dataset are located in Tables 14 and 15. Additional PARADE results for subgroup gaps excluding Fitzpatrick Skintone VI (as mentioned in Section 6.3) are located in Table 16.

			Overall					
			Contrastive · Distance		Margin · Distance		Margin · Semi-hard	
			Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced
Upstream Embedding		Recall@1	0.79 ± 0.002	0.782 ± 0.005	0.786 ± 0.003	0.78 ± 0.005	0.775 ± 0.006	0.766 ± 0.006
		NMI	0.872 ± 0.002	0.859 ± 0.003	0.861 ± 0.003	0.856 ± 0.004	0.856 ± 0.003	0.85 ± 0.005
		U_{KL}	0.397 ± 0.003	0.449 ± 0.007	0.076 ± 0.001	0.076 ± 0.001	0.113 ± 0.003	0.113 ± 0.002
Downstream Classification	LR	Accuracy	0.815 ± 0.002	0.81 ± 0.005	0.815 ± 0.002	0.827 ± 0.006	0.809 ± 0.004	0.818 ± 0.005
		Precision	0.817 ± 0.002	0.811 ± 0.005	0.822 ± 0.001	0.831 ± 0.006	0.815 ± 0.004	0.822 ± 0.005
		Recall	0.815 ± 0.002	0.81 ± 0.005	0.816 ± 0.002	0.827 ± 0.006	0.81 ± 0.004	0.818 ± 0.005
	RF	Accuracy	0.776 ± 0.002	0.787 ± 0.007	0.757 ± 0.003	0.735 ± 0.009	0.768 ± 0.002	0.758 ± 0.004
		Precision	0.785 ± 0.003	0.793 ± 0.006	0.762 ± 0.003	0.737 ± 0.01	0.774 ± 0.002	0.762 ± 0.004
		Recall	0.776 ± 0.002	0.787 ± 0.007	0.757 ± 0.003	0.735 ± 0.009	0.769 ± 0.002	0.758 ± 0.004
	SVM	Accuracy	0.813 ± 0.002	0.811 ± 0.007	0.81 ± 0.002	0.82 ± 0.007	0.808 ± 0.004	0.815 ± 0.004
		Precision	0.823 ± 0.003	0.821 ± 0.007	0.827 ± 0.002	0.843 ± 0.005	0.818 ± 0.003	0.829 ± 0.004
		Recall	0.813 ± 0.002	0.811 ± 0.007	0.811 ± 0.002	0.82 ± 0.006	0.808 ± 0.003	0.815 ± 0.004

			Overall							
			Multisimilarity		Proxy-NCA		Triplet · Distance		Triplet · Semi-hard	
			Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced
Upstream Embedding		Recall@1	0.779 ± 0.006	0.788 ± 0.005	0.807 ± 0.004	0.8 ± 0.007	0.792 ± 0.003	0.795 ± 0.007	0.761 ± 0.004	
		NMI	0.857 ± 0.003	0.857 ± 0.004	0.873 ± 0.003	0.86 ± 0.005	0.866 ± 0.002	0.861 ± 0.005	0.848 ± 0.005	0.843 ± 0.004
		U_{KL}	0.139 ± 0.001	0.146 ± 0.002	0.056 ± 0.001	0.073 ± 0.001	0.274 ± 0.005	0.277 ± 0.005	0.336 ± 0.004	0.321 ± 0.007
Downstream Classification	LR	Accuracy	0.813 ± 0.004	0.833 ± 0.004	0.824 ± 0.003	0.828 ± 0.005	0.82 ± 0.001	0.828 ± 0.005	0.802 ± 0.003	0.806 ± 0.004
		Precision	0.82 ± 0.004	0.836 ± 0.004	0.828 ± 0.003	0.833 ± 0.005	0.826 ± 0.001	0.833 ± 0.006	0.808 ± 0.003	0.811 ± 0.004
		Recall	0.814 ± 0.004	0.833 ± 0.004	0.824 ± 0.003	0.828 ± 0.005	0.82 ± 0.001	0.828 ± 0.005	0.803 ± 0.003	0.806 ± 0.005
	RF	Accuracy	0.754 ± 0.003	0.754 ± 0.005	0.761 ± 0.006	0.768 ± 0.007	0.786 ± 0.004	0.789 ± 0.008	0.776 ± 0.003	0.774 ± 0.007
		Precision	0.76 ± 0.004	0.755 ± 0.005	0.768 ± 0.007	0.774 ± 0.006	0.794 ± 0.004	0.792 ± 0.009	0.782 ± 0.003	0.778 ± 0.006
		Recall	0.754 ± 0.003	0.755 ± 0.005	0.762 ± 0.006	0.768 ± 0.007	0.786 ± 0.004	0.789 ± 0.008	0.777 ± 0.003	0.774 ± 0.007
	SVM	Accuracy	0.812 ± 0.002	0.828 ± 0.006	0.818 ± 0.002	0.816 ± 0.005	0.819 ± 0.002	0.83 ± 0.005	0.798 ± 0.001	0.808 ± 0.005
		Precision	0.825 ± 0.003	0.848 ± 0.005	0.834 ± 0.002	0.852 ± 0.004	0.829 ± 0.003	0.845 ± 0.004	0.806 ± 0.002	0.816 ± 0.005
		Recall	0.812 ± 0.002	0.828 ± 0.006	0.819 ± 0.002	0.817 ± 0.005	0.819 ± 0.002	0.831 ± 0.005	0.798 ± 0.001	0.808 ± 0.005

Table 9: *Overall results on CUB200.* Metrics over entire test dataset in representation space and downstream classification (LR, RF, and SVM) over 10 seed in manually class imbalanced experiments (Imbalanced) and control experiments (Balanced) for CUB200.

			Subgroup Gap					
			Contrastive · Distance		Margin · Distance		Margin · Semi-hard	
			Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced
Upstream Embedding		Recall@1	0.011 ± 0.004	0.168 ± 0.028	0.008 ± 0.005	0.212 ± 0.029	0.01 ± 0.008	0.187 ± 0.031
		NMI	-0.009 ± 0.002	0.109 ± 0.015	-0.008 ± 0.005	0.112 ± 0.012	-0.009 ± 0.003	0.092 ± 0.017
		U_{KL}	-0.112 ± 0.004	0.004 ± 0.011	-0.043 ± 0.002	0.0 ± 0.002	-0.05 ± 0.004	0.002 ± 0.004
Downstream Classification	LR	Accuracy	0.014 ± 0.004	0.181 ± 0.029	0.008 ± 0.003	0.131 ± 0.027	0.009 ± 0.006	0.131 ± 0.031
		Precision	0.333 ± 0.003	0.417 ± 0.012	0.337 ± 0.005	0.39 ± 0.014	0.331 ± 0.006	0.393 ± 0.015
		Recall	0.354 ± 0.004	0.462 ± 0.016	0.356 ± 0.005	0.424 ± 0.018	0.351 ± 0.007	0.43 ± 0.016
	RF	Accuracy	0.013 ± 0.004	0.121 ± 0.026	0.009 ± 0.005	0.325 ± 0.035	0.01 ± 0.006	0.255 ± 0.031
		Precision	0.339 ± 0.007	0.386 ± 0.014	0.347 ± 0.006	0.428 ± 0.011	0.342 ± 0.007	0.418 ± 0.014
		Recall	0.359 ± 0.006	0.391 ± 0.015	0.365 ± 0.006	0.495 ± 0.01	0.362 ± 0.007	0.478 ± 0.014
	SVM	Accuracy	0.014 ± 0.003	0.106 ± 0.032	0.009 ± 0.004	0.043 ± 0.028	0.009 ± 0.006	0.058 ± 0.029
		Precision	0.326 ± 0.008	0.36 ± 0.021	0.332 ± 0.008	0.301 ± 0.023	0.329 ± 0.006	0.329 ± 0.017
		Recall	0.345 ± 0.007	0.362 ± 0.024	0.348 ± 0.008	0.278 ± 0.027	0.348 ± 0.006	0.323 ± 0.02

			Subgroup Gap							
			Multisimilarity		Proxy-NCA		Triplet · Distance		Triplet · Semi-hard	
			Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced	Balanced	Imbalanced
Upstream Embedding		Recall@1	0.008 ± 0.009	0.187 ± 0.031	0.01 ± 0.005	0.256 ± 0.03	0.009 ± 0.004	0.159 ± 0.031	0.009 ± 0.006	0.168 ± 0.036
		NMI	-0.008 ± 0.004	0.113 ± 0.016	-0.009 ± 0.004	0.142 ± 0.015	-0.007 ± 0.003	0.103 ± 0.016	-0.01 ± 0.006	0.082 ± 0.016
		U_{KL}	-0.036 ± 0.002	-0.003 ± 0.003	-0.131 ± 0.003	-0.012 ± 0.005	-0.057 ± 0.006	-0.004 ± 0.009	-0.051 ± 0.005	0.014 ± 0.011
Downstream Classification	LR	Accuracy	0.009 ± 0.006	0.141 ± 0.032	0.007 ± 0.005	0.169 ± 0.027	0.011 ± 0.002	0.179 ± 0.031	0.011 ± 0.005	0.134 ± 0.031
		Precision	0.337 ± 0.008	0.391 ± 0.016	0.337 ± 0.005	0.428 ± 0.018	0.335 ± 0.005	0.41 ± 0.014	0.336 ± 0.006	0.384 ± 0.014
		Recall	0.356 ± 0.008	0.427 ± 0.019	0.356 ± 0.006	0.455 ± 0.019	0.355 ± 0.004	0.459 ± 0.016	0.357 ± 0.006	0.426 ± 0.016
	RF	Accuracy	0.009 ± 0.006	0.282 ± 0.034	0.009 ± 0.009	0.214 ± 0.026	0.01 ± 0.005	0.192 ± 0.035	0.011 ± 0.004	0.175 ± 0.03
		Precision	0.348 ± 0.006	0.428 ± 0.011	0.355 ± 0.01	0.436 ± 0.009	0.347 ± 0.006	0.409 ± 0.014	0.341 ± 0.007	0.393 ± 0.013
		Recall	0.365 ± 0.005	0.48 ± 0.012	0.372 ± 0.01	0.443 ± 0.01	0.364 ± 0.007	0.44 ± 0.014	0.36 ± 0.006	0.437 ± 0.015
	SVM	Accuracy	0.009 ± 0.003	0.048 ± 0.027	0.009 ± 0.003	0.063 ± 0.026	0.011 ± 0.003	0.071 ± 0.027	0.012 ± 0.002	0.082 ± 0.03
		Precision	0.335 ± 0.005	0.307 ± 0.02	0.33 ± 0.006	0.347 ± 0.022	0.334 ± 0.006	0.324 ± 0.014	0.334 ± 0.005	0.34 ± 0.012
		Recall	0.352 ± 0.004	0.284 ± 0.023	0.347 ± 0.004	0.299 ± 0.022	0.353 ± 0.006	0.315 ± 0.017	0.355 ± 0.005	0.356 ± 0.014

Table 10: *Gap study on CUB200.* Average gaps in representation space and downstream classification (LR, RF, and SVM) over 10 seeds between minoritized and majoritized classes in manually class imbalanced experiments (Imbalanced) and control experiments (Balanced) for CUB200.

C.4 LFW

Additional results for all loss and batch mining strategies for the LFW dataset are located in Tables 18 and 19. Per-subgroup results for LFW to demonstrate worst-group performance for the "White" subgroup (as mentioned in Section 6.3) are located in Table 20. Benchmarking of fairness improvement methods in downstream classification for bird color are shown in Table 21.

Table 11: *Benchmarking additional fairness improvement methods in downstream classification on CUB200 (Classes).* Overall performance and subgroup gaps for Domain-Independent Training and Oversampling (Wang et al., 2020b) on CUB200-2011 in class imbalanced experiments with upstream embedding trained on imbalanced dataset.

(a) Domain-Independent Training

METRIC ↓	Contr. (D)	Margin (D)	Margin (Sem.)	Msim.	ProxyNCA	Triplet (D)	Triplet (S)
Overall							
ACCURACY	0.782 ± 0.008	0.809 ± 0.004	0.794 ± 0.005	0.805 ± 0.004	0.823 ± 0.006	0.798 ± 0.005	0.749 ± 0.006
PRECISION	0.805 ± 0.011	0.840 ± 0.006	0.827 ± 0.005	0.847 ± 0.005	0.836 ± 0.006	0.842 ± 0.005	0.806 ± 0.006
RECALL	0.782 ± 0.008	0.809 ± 0.004	0.795 ± 0.005	0.805 ± 0.004	0.823 ± 0.006	0.798 ± 0.004	0.749 ± 0.005
Gap							
ACCURACY	0.034 ± 0.033	0.003 ± 0.030	0.014 ± 0.031	0.024 ± 0.031	0.140 ± 0.030	0.024 ± 0.030	0.077 ± 0.031
PRECISION	0.340 ± 0.024	0.304 ± 0.031	0.301 ± 0.022	0.264 ± 0.028	0.410 ± 0.022	0.262 ± 0.032	0.270 ± 0.018
RECALL	0.308 ± 0.027	0.253 ± 0.035	0.249 ± 0.025	0.189 ± 0.031	0.415 ± 0.024	0.188 ± 0.036	0.177 ± 0.021

(b) Oversampling

METRIC ↓	Contr. (D)	Margin (D)	Margin (Sem.)	Msim.	ProxyNCA	Triplet (D)	Triplet (S)
Overall							
ACCURACY	0.811 ± 0.005	0.828 ± 0.005	0.818 ± 0.004	0.832 ± 0.005	0.828 ± 0.005	0.829 ± 0.006	0.806 ± 0.005
PRECISION	0.814 ± 0.005	0.831 ± 0.005	0.822 ± 0.004	0.835 ± 0.005	0.833 ± 0.005	0.832 ± 0.006	0.811 ± 0.004
RECALL	0.812 ± 0.005	0.828 ± 0.005	0.819 ± 0.004	0.833 ± 0.005	0.828 ± 0.005	0.829 ± 0.006	0.807 ± 0.005
Gap							
ACCURACY	0.182 ± 0.027	0.131 ± 0.028	0.129 ± 0.030	0.142 ± 0.035	0.170 ± 0.026	0.177 ± 0.032	0.135 ± 0.032
PRECISION	0.421 ± 0.011	0.386 ± 0.015	0.391 ± 0.016	0.391 ± 0.016	0.428 ± 0.016	0.409 ± 0.015	0.385 ± 0.013
RECALL	0.464 ± 0.015	0.420 ± 0.018	0.428 ± 0.018	0.426 ± 0.020	0.455 ± 0.017	0.457 ± 0.017	0.427 ± 0.015

Table 12: *Benchmarking additional fairness improvement methods in downstream classification on CUB200 (Color).* Overall performance and subgroup gaps for Domain-Independent Training and Oversampling (Wang et al., 2020b) on CUB200-2011 in bird color experiments.

(a) Domain-Independent Training

METRIC ↓	Margin (D)
Overall	
ACCURACY	0.490 ± 0.005
PRECISION	0.896 ± 0.003
RECALL	0.489 ± 0.006
Gap	
ACCURACY	0.426 ± 0.017
PRECISION	0.185 ± 0.108
RECALL	0.353 ± 0.108

(b) Oversampling

METRIC ↓	Margin (D)
Overall	
ACCURACY	0.802 ± 0.002
PRECISION	0.816 ± 0.002
RECALL	0.802 ± 0.002
Gap	
ACCURACY	0.143 ± 0.019
PRECISION	0.323 ± 0.063
RECALL	0.348 ± 0.064

C.5 EXPLORATION OF FAIRNESS - UTILITY TRADEOFF AND VARYING HYPERPARAMETERS IN PARADE

We vary α_{SA} and ρ in the PARADE objective to explore the relationship between the overall performance, subgroup gap, and worst-group performance in PARADE. As stated in the main paper, we optimize α_{SA} and ρ via worst-group performance. Results of this analysis are displayed in Figure 12. We use our exploration to expound on how to optimize for α_{SA} and ρ. As seen in Figure 12, a clear trend that inversely relates overall performance, and fairness as measured by subgroup gap and worst-group performance is seen for the uniformity metric, U_{KL} over the grid of α_{SA} and ρ values (Note that higher values of U_{KL} correspond to *worse* performance). Recall@1 and NMI demonstrate noisier relationships between overall performance and fairness; and several α_{SA}, ρ choices appear to select an optimal tradeoff. In Figure 12, for Recall@1, we observe that at the location $\alpha_{SA} = 0.1$, $\rho = 500$. in the optimization grid, PARADE reaches peak overall performance *and* fairness (measured by low subgroup gap and high performance for the worst-performing subgroup) simultaneously. Thus, we could conclude that this choice of α_{SA} and ρ represents an optimal tradeoff for utility and fairness

Color	overall	overall	black	black	blue	blue	brown	brown
Method	Parade	Margin (D)	Parade	Margin (D)	Parade	Margin (D)	Parade	Margin (D)
Recall@1	0.785 ± 0.003	0.786 ± 0.003	0.780 ± 0.006	0.777 ± 0.005	0.837 ± 0.016	0.841 ± 0.010	0.773 ± 0.005	0.779 ± 0.008
NMI	0.860 ± 0.001	0.861 ± 0.003	0.832 ± 0.005	0.837 ± 0.004	0.840 ± 0.025	0.863 ± 0.026	0.831 ± 0.013	0.838 ± 0.007
U_{KL}	0.071 ± 0.002	0.076 ± 0.001	0.164 ± 0.002	0.153 ± 0.003	0.296 ± 0.004	0.275 ± 0.007	0.156 ± 0.004	0.148 ± 0.003
Precision	0.819 ± 0.003	0.822 ± 0.001	0.403 ± 0.007	0.412 ± 0.016	0.399 ± 0.053	0.401 ± 0.022	0.396 ± 0.005	0.390 ± 0.016
Recall	0.812 ± 0.003	0.816 ± 0.002	0.375 ± 0.007	0.384 ± 0.014	0.367 ± 0.052	0.366 ± 0.022	0.357 ± 0.008	0.351 ± 0.016
Accuracy	0.812 ± 0.003	0.815 ± 0.002	0.790 ± 0.009	0.798 ± 0.004	0.867 ± 0.009	0.877 ± 0.011	0.804 ± 0.001	0.812 ± 0.006

Color	buff	buff	green	green	grey	grey	iridescent	iridescent	olive
Method	Parade	Margin (D)	Parade	Margin (D)	Parade	Margin (D)	Parade	Margin (D)	Parade
Recall@1	0.787 ± 0.008	0.792 ± 0.006	1.000 ± 0.000	1.000 ± 0.000	0.751 ± 0.015	0.754 ± 0.010	1.000 ± 0.000	1.000 ± 0.000	0.589 ± 0.038
NMI	0.789 ± 0.005	0.808 ± 0.015	-0.000 ± 0.000	-0.000 ± 0.000	0.853 ± 0.009	0.849 ± 0.004	1.000 ± 0.000	0.800 ± 0.447	-0.000 ± 0.000
U_{KL}	0.349 ± 0.003	0.343 ± 0.004	0.262 ± 0.024	0.252 ± 0.021	0.142 ± 0.005	0.137 ± 0.002	0.369 ± 0.013	0.318 ± 0.017	0.164 ± 0.008
Precision	0.250 ± 0.006	0.253 ± 0.013	1.000 ± 0.000	1.000 ± 0.000	0.337 ± 0.014	0.338 ± 0.022	1.000 ± 0.000	1.000 ± 0.000	0.289 ± 0.077
Recall	0.209 ± 0.005	0.212 ± 0.012	1.000 ± 0.000	1.000 ± 0.000	0.292 ± 0.013	0.297 ± 0.021	1.000 ± 0.000	1.000 ± 0.000	0.173 ± 0.048
Accuracy	0.824 ± 0.003	0.824 ± 0.011	1.000 ± 0.000	1.000 ± 0.000	0.790 ± 0.007	0.796 ± 0.006	1.000 ± 0.000	1.000 ± 0.000	0.600 ± 0.033

Color	olive	orange	orange	red	red	white	white	yellow	yellow
Method	Margin (D)	Parade	Margin (D)	Parade	Margin (D)	Parade	Margin (D)	Parade	Margin (D)
Recall@1	0.620 ± 0.087	0.839 ± 0.010	0.863 ± 0.040	0.917 ± 0.018	0.919 ± 0.017	0.729 ± 0.012	0.706 ± 0.011	0.846 ± 0.003	0.862 ± 0.012
NMI	0.000 ± 0.000	0.701 ± 0.060	0.657 ± 0.036	0.839 ± 0.032	0.830 ± 0.033	0.792 ± 0.008	0.787 ± 0.010	0.842 ± 0.011	0.864 ± 0.005
U_{KL}	0.151 ± 0.005	0.295 ± 0.004	0.277 ± 0.009	0.445 ± 0.003	0.411 ± 0.011	0.176 ± 0.003	0.166 ± 0.003	0.215 ± 0.005	0.202 ± 0.004
Precision	0.240 ± 0.063	0.302 ± 0.027	0.303 ± 0.079	0.555 ± 0.052	0.559 ± 0.048	0.387 ± 0.018	0.380 ± 0.012	0.503 ± 0.012	0.530 ± 0.022
Recall	0.160 ± 0.049	0.261 ± 0.024	0.263 ± 0.076	0.542 ± 0.053	0.544 ± 0.049	0.357 ± 0.018	0.342 ± 0.011	0.481 ± 0.011	0.509 ± 0.022
Accuracy	0.660 ± 0.060	0.867 ± 0.017	0.860 ± 0.028	0.923 ± 0.017	0.927 ± 0.009	0.752 ± 0.004	0.737 ± 0.002	0.879 ± 0.001	0.884 ± 0.008

Table 13: *Absolute performance for all CUB200 subgroups.* Metrics over each bird color subgroup in the CUB200 test dataset respectively, in representation space and downstream classification (logistic regressor) over 3 seeds for standard methods and PARADE in CUB200.

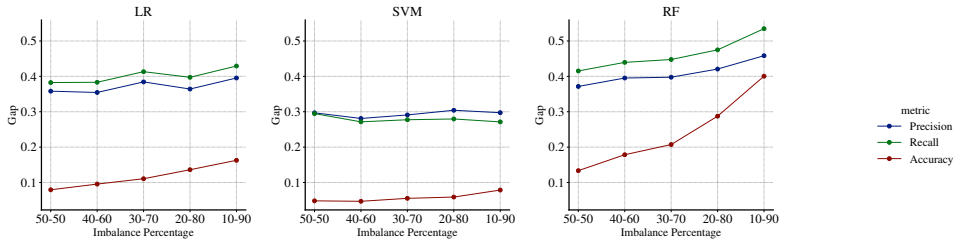


Figure 9: Impact of varying imbalance between the *minoritized* and *majoritized* classes on various downstream classifiers (RF, LR and SVM) in the manually class imbalanced CUB200 experiments. (Note: the imbalance percentage $50 - 50$ is equivalent to the balanced setting). Gaps increase for all classifiers downstream with more imbalance introduced to the upstream training data.

			Arcface		Overall Margin · Distance		N-Pair · N-Pair
			PARADE	Standard	PARADE	Standard	Standard
UPSTREAM EMBEDDING		Recall@1	0.897 ± 0.002	0.888 ± 0.002	0.885 ± 0.002	0.922 ± 0.001	0.11 ± 0.002
		NMI	0.91 ± 0.0	0.902 ± 0.001	0.901 ± 0.003	0.929 ± 0.0	0.61 ± 0.0
		U_{KL}	0.019 ± 0.001	0.017 ± 0.0	0.336 ± 0.013	0.237 ± 0.006	2.595 ± 0.055
DOWNSTREAM CLASSIFICATION	LR	Accuracy	0.891 ± 0.0	0.89 ± 0.001	0.692 ± 0.004	0.831 ± 0.002	0.017 ± 0.0
		Precision	0.721 ± 0.0	0.721 ± 0.001	0.55 ± 0.003	0.652 ± 0.003	0.004 ± 0.0
		Recall	0.74 ± 0.0	0.741 ± 0.001	0.546 ± 0.003	0.674 ± 0.003	0.011 ± 0.0

Table 14: *Overall results on CelebA.* Metrics over entire test dataset in representation space and downstream classification (logistic regressor) over 3 seeds for standard methods and PARADE in CelebA.

in PARADE as measured by Recall@1. By the other displayed metrics, we see that $\alpha_{SA} = 0.1$, $\rho = 500$. demonstrates a reasonable utility-fairness tradeoff. Therefore, the choice of $\alpha_{SA} = 0.1$, $\rho = 500$. would be optimal for PARADE in CUB200 bird color setting. Note that the choice of where to operate within this trade-off should depend on the application that is being targeted. For

			Subgroup Gap				
			Arcface		Margin · Distance		N-Pair · N-Pair
			PARADE	Standard	PARADE	Standard	Standard
UPSTREAM EMBEDDING		Recall@1	0.135 ± 0.008	0.128 ± 0.003	0.085 ± 0.009	0.122 ± 0.005	-0.023 ± 0.013
		NMI	-0.003 ± 0.004	-0.01 ± 0.002	-0.012 ± 0.003	-0.002 ± 0.003	-0.102 ± 0.002
		U_{KL}	-0.054 ± 0.003	-0.052 ± 0.003	-0.04 ± 0.011	-0.03 ± 0.007	-0.015 ± 0.038
DOWNSTREAM CLASSIFICATION	LR	Accuracy	0.068 ± 0.002	0.069 ± 0.002	0.131 ± 0.006	0.082 ± 0.005	0.006 ± 0.002
		Precision	0.087 ± 0.003	0.087 ± 0.004	0.146 ± 0.006	0.1 ± 0.007	0.001 ± 0.001
		Recall	0.084 ± 0.002	0.083 ± 0.003	0.141 ± 0.007	0.098 ± 0.007	0.002 ± 0.001

Table 15: *Gap study on CelebA.* Average gaps in representation space and downstream classification (logistic regressor) over 3 seeds between minoritized and majoritized classes (Fitzpatrick Skintone) for standard methods and PARADE in CelebA.

		Margin · Distance	
		PARADE	Standard
UPSTREAM EMBEDDING	Recall@1	-0.035 ± 0.006	0.005 ± 0.004
	NMI	-0.004 ± 0.003	0.004 ± 0.002
	U_{KL}	0.04 ± 0.011	0.084 ± 0.006
DOWNSTREAM CLASSIFICATION (LR)	Precision	0.021 ± 0.006	0.039 ± 0.002
	Recall	0.018 ± 0.006	0.029 ± 0.002
	Accuracy	0.011 ± 0.005	0.018 ± 0.002

Table 16: *Gap study on CelebA excluding Fitzpatrick Skintone VI.* Average gaps in representation space and downstream classification (logistic regressor) over 3 seeds between minoritized and majoritized classes (Fitzpatrick Skintone) where the darkest skintone (VI) is excluded for standard methods and PARADE in CelebA.

Skintones	Overall	Overall	Skintone 1	Skintone 1	Skintone 2	Skintone 2	Skintone 3
Method	Parade	Margin (D)	Parade	Margin (D)	Parade	Margin (D)	Parade
Recall@1	0.885 ± 0.002	0.922 ± 0.001	0.738 ± 0.005	0.858 ± 0.011	0.887 ± 0.004	0.930 ± 0.001	0.907 ± 0.001
NMI@1	0.901 ± 0.003	0.929 ± 0.000	0.933 ± 0.007	0.961 ± 0.000	0.923 ± 0.004	0.947 ± 0.001	0.927 ± 0.002
U_{KL}	0.336 ± 0.013	0.237 ± 0.006	0.436 ± 0.020	0.419 ± 0.002	0.350 ± 0.014	0.260 ± 0.004	0.350 ± 0.012
Precision	0.550 ± 0.003	0.652 ± 0.003	0.398 ± 0.011	0.639 ± 0.005	0.566 ± 0.003	0.696 ± 0.002	0.558 ± 0.003
Recall	0.546 ± 0.003	0.674 ± 0.003	0.421 ± 0.014	0.656 ± 0.005	0.604 ± 0.003	0.739 ± 0.003	0.578 ± 0.003
Accuracy	0.692 ± 0.004	0.831 ± 0.002	0.578 ± 0.009	0.783 ± 0.004	0.707 ± 0.004	0.843 ± 0.002	0.716 ± 0.004
Skintones	Skintone 3	Skintone 4	Skintone 4	Skintone 5	Skintone 5	Skintone 6	Skintone 6
Method	Margin (D)	Parade	Margin (D)	Parade	Margin (D)	Parade	Margin (D)
Recall@1	0.937 ± 0.003	0.850 ± 0.002	0.893 ± 0.003	0.785 ± 0.017	0.838 ± 0.004	0.642 ± 0.018	0.628 ± 0.006
NMI@1	0.947 ± 0.001	0.927 ± 0.002	0.946 ± 0.000	0.943 ± 0.002	0.957 ± 0.004	0.948 ± 0.004	0.960 ± 0.009
U_{KL}	0.241 ± 0.006	0.339 ± 0.012	0.240 ± 0.008	0.371 ± 0.012	0.288 ± 0.013	0.547 ± 0.010	0.483 ± 0.014
Precision	0.672 ± 0.003	0.471 ± 0.005	0.644 ± 0.003	0.355 ± 0.011	0.570 ± 0.001	0.258 ± 0.005	0.492 ± 0.021
Recall	0.708 ± 0.003	0.518 ± 0.005	0.695 ± 0.002	0.386 ± 0.011	0.602 ± 0.001	0.275 ± 0.005	0.511 ± 0.019
Accuracy	0.842 ± 0.003	0.632 ± 0.005	0.798 ± 0.002	0.545 ± 0.009	0.747 ± 0.002	0.430 ± 0.010	0.678 ± 0.015

Table 17: *Absolute performance for all CelebA subgroups.* Metrics over each Fitzpatrick Skintone subgroup in the CelebA test dataset respectively, in representation space and downstream classification (logistic regressor) over 3 seeds for standard methods and PARADE in CelebA.

example, here we use Recall@1 to determine the optimal choice of hyperparameters and validate with the other two considered metrics. However, for LFW, which has a high population of singleton classes (see Figure 7), NMI would be a better metric to use for selecting optimal point.

D IMPLEMENTATION DETAILS

D.1 DATASET ATTRIBUTE INFORMATION

Dataset manipulation for the CARS196 and CUB200 manually class imbalanced experimentsis explained in Section 3.3.

Figure 10: A t-SNE (Maaten & Hinton, 2008) visualization of the two distinct PARADE embeddings for Fitzpatrick Skintone CelebA experiments: the sensitive attribute embedding (**left**) and the class label embedding (**right**).

			Overall				
			Arcface		Margin · Distance		N-Pair · N-Pair
			PARADE	Standard	PARADE	Standard	Standard
Upstream Embedding		Recall@1	0.268 ± 0.01	0.306 ± 0.008	0.329 ± 0.002	0.381 ± 0.004	0.187 ± 0.006
		NMI	0.849 ± 0.005	0.859 ± 0.002	0.865 ± 0.001	0.869 ± 0.001	0.854 ± 0.001
		U_{KL}	0.118 ± 0.021	0.089 ± 0.014	0.129 ± 0.001	0.103 ± 0.001	1.815 ± 0.011
Downstream Classification	RF	Accuracy	0.8 ± 0.0	0.804 ± 0.017	0.762 ± 0.002	0.8 ± 0.003	0.887 ± 0.002
		Precision	0.789 ± 0.003	0.793 ± 0.016	0.767 ± 0.001	0.788 ± 0.004	0.878 ± 0.003
		Recall	0.827 ± 0.0	0.831 ± 0.015	0.801 ± 0.003	0.823 ± 0.003	0.913 ± 0.002

Table 18: *Overall results on LFW.* Metrics over entire test dataset in representation space and downstream classification (random forest) over 3 seeds for standard methods and PARADE in LFW. **Note**: Due to the number of singleton classes in LFW, Recall@1 is not considered a good metric of performance for this dataset.

			Subgroup Gap				
			Arcface		Margin · Distance		N-Pair · N-Pair
			PARADE	Standard	PARADE	Standard	Standard
Upstream Embedding		Recall@1	0.039 ± 0.017	0.061 ± 0.017	0.075 ± 0.014	0.068 ± 0.013	0.054 ± 0.01
		NMI	0.048 ± 0.011	0.057 ± 0.003	0.041 ± 0.003	0.048 ± 0.003	0.048 ± 0.003
		U_{KL}	0.176 ± 0.019	0.157 ± 0.011	0.163 ± 0.003	0.165 ± 0.005	0.357 ± 0.012
Downstream Classification	RF	Accuracy	0.04 ± 0.01	0.038 ± 0.012	0.049 ± 0.005	0.038 ± 0.005	0.025 ± 0.004
		Precision	0.036 ± 0.018	0.041 ± 0.014	0.04 ± 0.005	0.037 ± 0.007	0.025 ± 0.007
		Recall	0.066 ± 0.017	0.076 ± 0.015	0.066 ± 0.006	0.071 ± 0.007	0.041 ± 0.006

Table 19: *Gap study on LFW.* Average gaps in representation space and downstream classification (random forest) over 3 seeds between minoritized and majoritized classes (Race) for standard methods and PARADE in LFW.

			Asian		Black		Indian		White	
			Margin · Distance							
			PARADE	Standard	PARADE	Standard	PARADE	Standard	PARADE	Standard
Upstream Embedding		Recall@1	0.262 ± 0.028	0.31 ± 0.014	0.289 ± 0.011	0.331 ± 0.016	0.238 ± 0.027	0.325 ± 0.032	0.338 ± 0.004	0.39 ± 0.004
		NMI	0.894 ± 0.003	0.914 ± 0.005	0.858 ± 0.001	0.882 ± 0.005	0.948 ± 0.007	0.951 ± 0.007	0.862 ± 0.001	0.868 ± 0.0
		U_{KL}	0.304 ± 0.003	0.265 ± 0.003	0.456 ± 0.009	0.417 ± 0.013	0.141 ± 0.002	0.133 ± 0.005	0.137 ± 0.002	0.107 ± 0.001
Downstream Classification	RF	Accuracy	0.81 ± 0.007	0.828 ± 0.008	0.827 ± 0.005	0.853 ± 0.007	0.772 ± 0.011	0.814 ± 0.012	0.754 ± 0.002	0.794 ± 0.003
		Precision	0.715 ± 0.012	0.726 ± 0.013	0.743 ± 0.008	0.759 ± 0.014	0.664 ± 0.006	0.711 ± 0.01	0.747 ± 0.002	0.769 ± 0.004
		Recall	0.713 ± 0.013	0.72 ± 0.014	0.751 ± 0.008	0.758 ± 0.01	0.657 ± 0.008	0.702 ± 0.011	0.773 ± 0.002	0.798 ± 0.003

Table 20: *Absolute performance for all LFW subgroups.* Metrics over each Race subgroup in the LFW test dataset respectively, in representation space and downstream classification (random forest) over 3 seeds for standard methods and PARADE in LFW.

For CUB200 bird color experiments, we utilized the labeled bird color attributes from Wah et al. (2011). Each image can have multiple "primary color" labels. Therefore, we take the mode over

Table 21: *Benchmarking additional fairness improvement methods in downstream classification on LFW.* Overall performance and subgroup gaps for Domain-Independent Training and Oversampling (Wang et al., 2020b) on LFW with Race attribute.

(a) Domain-Independent Training

METRIC ↓	ArcFace	Margin (D)	N-Pair
Overall			
ACCURACY	0.759 ± 0.017	0.753 ± 0.002	0.861 ± 0.005
PRECISION	0.793 ± 0.010	0.786 ± 0.003	0.872 ± 0.003
RECALL	0.799 ± 0.012	0.792 ± 0.003	0.889 ± 0.003
Gap			
ACCURACY	0.093 ± 0.011	0.095 ± 0.006	0.029 ± 0.009
PRECISION	0.030 ± 0.011	0.020 ± 0.009	0.029 ± 0.010
RECALL	0.040 ± 0.012	0.044 ± 0.008	0.026 ± 0.010

(b) Oversampling

METRIC ↓	ArcFace	Margin (D)	N-Pair
Overall			
ACCURACY	0.775 ± 0.017	0.767 ± 0.004	0.881 ± 0.004
PRECISION	0.771 ± 0.014	0.762 ± 0.002	0.873 ± 0.005
RECALL	0.815 ± 0.014	0.807 ± 0.002	0.909 ± 0.004
Gap			
ACCURACY	0.070 ± 0.012	0.072 ± 0.006	0.032 ± 0.006
PRECISION	0.020 ± 0.013	0.024 ± 0.009	0.027 ± 0.009
RECALL	0.023 ± 0.013	0.031 ± 0.011	0.036 ± 0.008

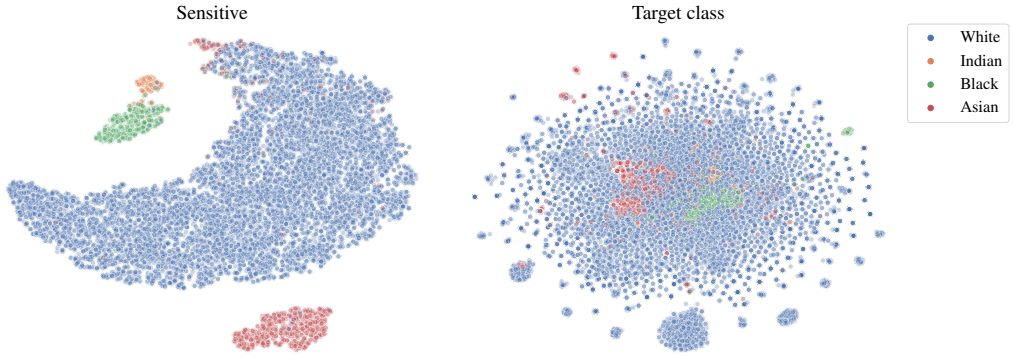


Figure 11: A t-SNE (Maaten & Hinton, 2008) visualization of the two distinct PARADE embeddings for Race LFW experiments: the sensitive attribute embedding (**left**) and the class label embedding (**right**).

Dataset	Protected Attribute	Protected Attribute Values
CUB200-2011	Color	Black, Blue, Brown, Buff, Green, Grey, Iridescent Olive, Orange, Red, White, Yellow
CelebA	Fitzpatrick Skintone Category	I, II, III, IV, V, VI
LFW	Race	Asian, Black, Indian, White

Table 22: *Summarizing attribute information.* Protected attribute examined and associated values taken by the protected attribute in each dataset analyzed w.r.t. a sensitive attribute in the main paper (CUB200, CelebA, LFW).

all bird colors labeled for each image in order to determine a single bird color associated with the image. For CelebA, we calculate the Fitzpatrick skintone based on the image pixel information for each image. The calculation is described in Section D.2. For LFW, we construct the "Race" attribute from labels of "White", "Black", "Asian," and "Indian" as labelled by Kumar et al. (2009). For each of these attributes, the labelling provided by Kumar et al. (2009) has a float value, which we map to binary values: the image is considered to have the attribute if the value is greater than 0, and the image is considered to not have the attribute if the value is less than 0. Naturally, the labelling is not necessarily correct for each image, as the confidence about the "Race" labelling can be quite low for some images. We remove all images without at least one of these attributes, though we note that these attributes do not encompass all races. Therefore, our analysis may not be relevant for other races not labelled by Kumar et al. (2009).

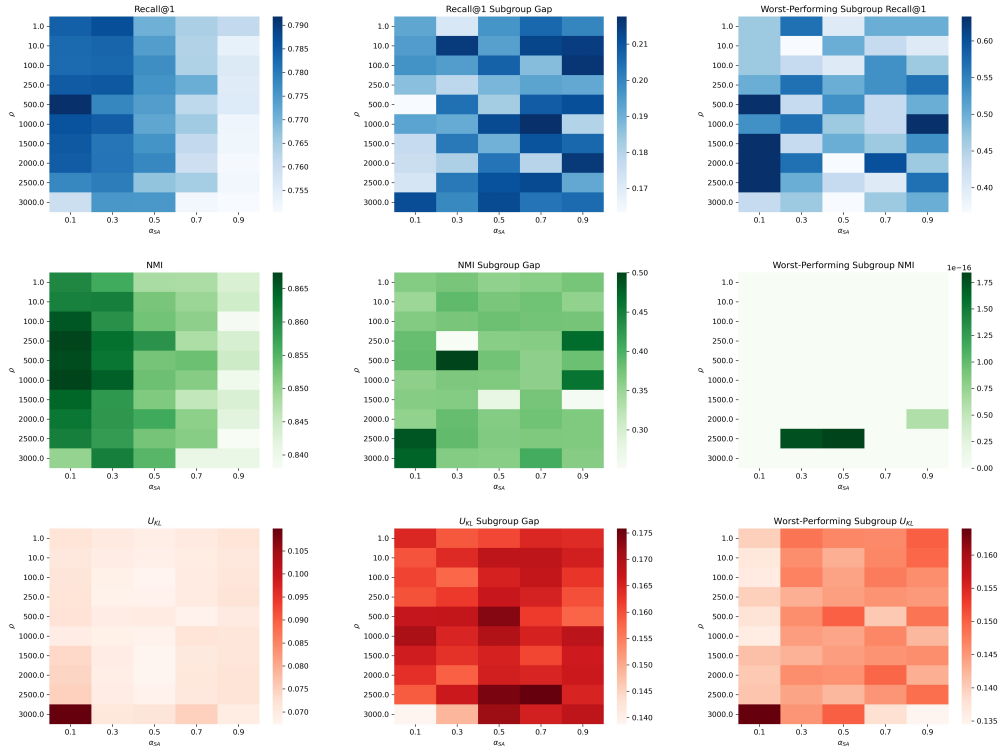


Figure 12: *Exploring fairness-utility tradeoffs in PARADE on CUB200 over grid of α_{SA} and ρ values.* Overall performance (left column), subgroup gap (middle column) and worst-group performance (right column) over metrics Recall@1 (top row), NMI (middle row), and U_{KL} (bottom row) in PARADE on CUB200. α_{SA} and ρ in PARADE objective (Section 4) varied from 0.1 to 0.9, and 1 to 3000, respectively.

D.2 Fitzpatrick Skintone Calculation

We follow the methods from Cheng et al. (2021) for calculation of Fitzpatrick Skintone based on image pixel information. However, we calculate these values for CelebA, as opposed to CelebA-HQ. As CelebA-HQ incorporates higher resolution images, but has fewer images, our process of Fitzpatrick Skintone calculation on CelebA is slightly modified to account for lower resolution, and differing image size.

In Cheng et al. (2021), two sample skin patches are selected from each image of CelebA-HQ to determine the skintone. We select three sample skin patches, as we are forced to reduce the dimensions of the patches to account for the smaller image size of CelebA. Additionally, we leverage facial landmark attributes provided by CelebA Liu et al. (2015) in order to choose our sample patches. Specifically, given the (x, y) landmarks for the left eye, right eye, and nose for each image, we choose to sample square patches of size 20×20 (all 3 color channels are selected) with the following center points:

$$(x_{\text{left eye}}, y_{\text{nose}})$$
$$(x_{\text{right eye}}, y_{\text{nose}})$$
$$(x_{\text{nose}}, y_{\text{nose}})$$

The first two center points are intended to capture the likely location of the *left* and *right* cheeks, respectively, as these are likely located below each eye and adjacent to the nose. The last center point is the nose. We note that this protected attribute generation is not perfect. In some cases, such label generation can accidentally use aspects of the background, if, the individual's face position in the image is not facing forward. Also, extreme lighting can lead to misclassification of skintone. Nonetheless, we believe the procedure provides a good approximation of Fitzpatrick skintone category, but do not recommend these attribute labels for use outside of fairness analysis.

The selected sample patches are converted to CIELab-space to retrieve the L (luminance) and b (yellow) values. We then calculate the Mean Individual Typology Angle (ITA) value:

$$ITA = \arctan\left(\frac{L-50}{b}\right) \times \frac{180^\circ}{\pi}$$

Table 23: Fitzpatrick Skin Tone Categories corresponding to Mean ITA values, information taken from Cheng et al. (2021)

ITA Range	Fitzpatrick Category	Description
50 $\leq$ ITA	I	Extremely Light
40 $\leq$ ITA < 50	II	Very Light
30 $\leq$ ITA < 40	III	Light / Somewhat Light
20 $\leq$ ITA < 30	IV	Dark / Somewhat Dark
10 $\leq$ ITA < 20	V	Very Dark
ITA < 10	VI	Extremely Dark

Based on the Mean ITA calculation, we classify each image into one of the 6 Fitzpatrick skintone categories, as listed in Table 23. To calculate subgroup gaps, we calculate gaps between the mean value over the 3 lightest Fitzpatrick skintones and the mean value over the 3 darkest Fitzpatrick skintones.

D.3 Training Parameters

For CUB200 and CARS196, we did not perform hyperparameter search but followed reported hyperparameters from Roth et al. (2020c) for best performance with an ImageNet Deng et al. (2009) pretrained ResNet50 He et al. (2016) and frozen batch normalization layers. As detailed in Roth et al. (2020c), we train for 150 epochs with embedding dimension 128, learning rate 0.00001 with no scheduler, and weight decay 0.0004. We train with a batch size of 128, with the Adam optimizer Kingma & Ba (2015) over five seeds inclusive for the balance control datasets, and for CUB200 color experiments; and seeds $0-9$ for the manually class imbalanced experiments. For training transforms, we normalize each image using color channel means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$, randomly crop the image and re-size to 224×224 and horizontally flip with probability 0.5. For testing transforms, we normalize each image with the aforementioned color channel means and standard deviations, resize to 256×256, and center crop to 224×224.

For CelebA and LFW, we performed hyperparameter search over the following hyperparameters: **architectures**: ResNet50 He et al. (2016), and SE-Net50 (both with and without frozen batch normalization layers); **number of training epochs**; **learning rates**; **last linear layer learning rate** (differ from other layer learning rates); **learning rate schedulers**; **embedding dimensions**: 64, 128, 256; **pre-training**; **image augmentations**. We evaluated hyperparameter sets on a validation set we cut from the typical training set (20% of training set), and chose the set of hyperparameters with best recall@k score for CelebA and best NMI score for LFW. NMI is used for LFW due to the high number of singleton classes present in the dataset (recall@1 is meaningless for singleton classes).

For CelebA, we train on the ResNet50 He et al. (2016) architecture with frozen batch normalization layers, for 125 epochs with learning rate 0.00001, and no scheduler, weight decay 0.0004, Adam Kingma & Ba (2015) optimizer, and batch size of 128. For training transforms, we normalize each image using color channel means $(0.5, 0.5, 0.5)$ and standard deviations $(0.5, 0.5, 0.5)$, resize to 256×256, center crop to 224×224 and horizontally flip with probability 0.5. For testing transforms, we normalize each image with the aforementioned color channel means and standard deviations, resize to 256×256, and center crop to 224×224. We average over runs with seeds $0-2$, inclusive.

For LFW, we train on the ResNet50 He et al. (2016) architecture with frozen batch normalization layers, for 125 epochs with initial learning rate 0.00001 for all model parameters except the last linear layer, which has initial learning rate 0.0001, and a multi-step learning rate scheduler which reduces the learning rate by a factor of 0.3 at epochs 50 and 100, weight decay 0.0004, Adam Kingma & Ba

(2015) optimizer, and batch size of 64. For training transforms, we normalize each image using color channel means $(0.5, 0.5, 0.5)$ and standard deviations $(0.5, 0.5, 0.5)$, resize to 256×256, center crop to 224×224 and horizontally flip with probability 0.5. For testing transforms, we normalize each image with the aforementioned color channel means and standard deviations, resize to 256×256, and center crop to 224×224. We average over runs with seeds $0 - 2$, inclusive.

For each dataset we chose a set of loss and batch mining strategies that have historically been used for the relevant task, encompassing a broad range of methods, and / or achieved good performance. However, for n-pair loss and sampling, good performance was not achieved for the facial datasets despite use in the past for facial recognition Sohn (2016). For manually class imbalanced experiments with CARS196 and CUB200 and the associated balanced controls, we used: margin loss / distance-weighted sampling, margin loss / semi-hard sampling, triplet loss / distance-weighted sampling, triplet loss / semi-hard sampling, contrastive loss / distance-weighted sampling, multisimilarity loss, and proxy-NCA loss. For the color experiments with CUB200, we used: margin loss / distance-weighted sampling. For CelebA and LFW, we used: margin loss / distance-weighted sampling, arcface loss, and n-pair loss and sampling. For all testing and evaluation experiments with PARADE, we used margin loss and distance-weighted sampling, but PARADE can be used with any loss and mining strategy.

DML-specific parameters Here we list the hyperparameters that we use for each evaluated loss function and batch mining strategy, if applicable. Refer to A for explicit formulas associated with the parameters here. We set $\gamma = 0.2$ in semi-hard mining. For distance-weighted mining, we set $\lambda = 0.5$ and clip the maximum distance to 1.4. In the triplet objective, we use $\gamma = 0.2$ for triplet loss. For margin loss, the learning rate of the boundary β is set to 0.0005, with initial value 1.2 and triplet margin $\gamma = 0.2$. For N-Pair uses embedding regularization parameter $\nu = 0.005$. In Multisimilarity loss, we use $\alpha = 2$, $\beta = 40$, $\lambda = 0.5$ and $\epsilon = 0.1$. Finally, for ArcFace, additive angular margin penalty is set to $\gamma = 0.5$, while scaling parameter $s = 16$ and class centers are optimized with learning rate 0.0005.

The two PARADE parameters, α_{SA} and ρ, as described in Section 4, were optimized via worst-group performance over a grid search. For CUB200, we set $\alpha_{SA} = 0.3$, $\rho = 1500$. For CelebA, we set $\alpha_{SA} = 0.1$, $\rho = 1000$. For LFW, we set $\alpha_{SA} = 0.3$, $\rho = 100$.

D.4 Fairness Evaluation

For each dataset, we calculate subgroup gaps between the *majoritized* and *minoritized* subgroup (CARS196, CUB200 *class*, CelebA) or between the worst-performing subgroup and other subgroups (LFW). In CUB200 *color* experiments, due to the large number of subgroups, we calculate the gap between the top 6 performing subgroups and the bottom 6 performing subgroups (there are 12 total subgroups).

Upstream In the upstream embedding tasks, in which we denote ϕ as the embedding function for the learned model, and use $A(x)$ to denote the value of the attribute A for data point x, we calculate recall@1 for data samples in X with associated class label Y and attribute $a \in A$ as:

$$\text{Recall@k} = \frac{1}{|\{x \in X : A(x) = a\}|} \sum_{\{x \in X : A(x) = a\}} \begin{cases} 1 & \exists \tilde{x} \in NN_k(x) : Y(\tilde{x}) = Y(x) \\ 0 & \text{else} \end{cases}$$

Note here that the nearest neighbors function is computed with respect to *all* $x \in X$, not exclusively $x \in X$ with attribute $a \in A$, but the input to the nearest neighbors function is exclusively $\{x \in X : A(x) = a\}$. To calculate NMI, let $\mathcal{C}$ be the output of a clustering algorithm C on the entire dataset X, i.e. $\mathcal{C} = C(X)$ and let $\mathcal{C}|_{\mathcal{S}}$ denote the output of clustering algorithm C restricted to some subset $\mathcal{S} \subset X$. The important note here is that the clustering algorithm is run over the entire dataset, but $\mathcal{C}|_{\mathcal{S}}$ expresses the cluster labels only for $\mathcal{S} \subset X$. Then, we measure NMI for data samples in X with associated class label Y and attribute $a \in A$ as:

$$NMI = \frac{I(Y(\{x \in X : A(x) = a\}); \mathcal{C}|_{\{x \in X : A(x) = a\}})}{H(Y(\{x \in X : A(x) = a\})) + H(\mathcal{C}|_{\{x \in X : A(x) = a\}})}$$

We calculate U_{KL} for data samples in X with attribute $a \in A$ as:

$$U_{\text{KL}}(X) = \mathcal{D}_{\text{KL}}\left(\mathcal{U}_{\text{D}}, \mathcal{S}_{\phi(\{x \in X : A(x) = a\})}\right)$$

where $\mathcal{S}_{\phi(\{x \in X : A(x)=a\})}$ denotes the singular values over $\phi(\{x \in X : A(x) = a\})$.

Downstream In the downstream tasks, for data samples in X with class label Y, and predictor $\hat{Y}$, let $Y(x)$ express the value of the ground-truth label for data sample x and let $\hat{Y}(x)$ express the value of the predicted label. Then, we denote $TP_a^{(y)}$ the number of true positives with attribute $a \in A$:

$$TP_a^{(y)} = \{x \in X : A(x) = a, Y(x) = y, \hat{Y}(x) = y\}$$

$FP_a^{(y)}$ the number of false positives with attribute $a \in A$

$$FP_a^{(y)} = \{x \in X : A(x) = a, Y(x) = y, \hat{Y}(x) \neq y\}$$

and $FN_a^{(y)}$ the number of false negatives with attribute $a \in A$:

$$FN_a^{(y)} = \{x \in X : A(x) = a, Y(x) \neq y, \hat{Y}(x) = y\}$$

for $y \in Y$.

We calculate macro-averaged recall for data samples in X with associated class label Y and attribute $a \in A$ as:

$$\text{Recall} = \frac{1}{|Y|} \sum_{y \in Y} \frac{TP_a^{(y)}}{TP_a^{(y)} + FN_a^{(y)}}$$

where $|Y|$ is the number of possible class labels, i.e. the size of the set of all possible values of Y. We calculate macro-averaged precision for data samples in X with associated class label Y and attribute $a \in A$ as:

$$\text{Precision} = \frac{1}{|Y|} \sum_{y \in Y} \frac{TP_a^{(y)}}{TP_a^{(y)} + FP_a^{(y)}}$$

We calculate accuracy for data samples in X with associated class label Y and attribute $a \in A$ as:

$$\text{Accuracy} = \frac{|\{x \in X : A(x) = a, Y(x) = \hat{Y}(x)\}|}{|\{x \in X : A(x) = a\}|}$$

The subgroup gaps are then considered to be the difference between the metric value for the *majoritized* subgroup and the metric value for the *minoritized* subgroup (CARS196, CUB200, CelebA); or between the metric value for *each subgroup with better performance than the worst-performing subgroup* and the metric value for the *worst-performing subgroup* (LFW). As stated in Section D.4, for CUB200 bird color experiments, the subgroup gaps were calculated between the top performing 50% of subgroups and bottom performing 50 of subgroups.

FAIRNESS GUARANTEES UNDER DEMOGRAPHIC SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies found that using machine learning for social applications can lead to injustice in the form of racist, sexist, and otherwise unfair and discriminatory outcomes. To address this challenge, recent machine learning algorithms have been designed to limit the likelihood such unfair behavior occurs. However, these approaches typically assume the data used for training is representative of what will be encountered in deployment, which is often untrue. In particular, if certain subgroups of the population become more or less probable in deployment (a phenomenon we call *demographic shift*), prior work's fairness assurances are often invalid. In this paper, we consider the impact of demographic shift and present a class of algorithms, called `Shifty` algorithms, that provide high-confidence behavioral guarantees that hold under demographic shift. `Shifty`, the first technique of its kind, demonstrates an effective strategy for designing algorithms to overcome demographic shift's challenges. We evaluate `Shifty` using a real-world dataset of university entrance exams and subsequent student success. We show that the learned models avoid bias under demographic shift, unlike existing methods. Our experiments demonstrate that our algorithm's high-confidence fairness guarantees are valid in practice and that our algorithm is an effective tool for training models that are fair when demographic shift occurs.

1 INTRODUCTION

As machine learning (ML) algorithms continue to be used to aid decisions in socially impactful applications (Angwin et al., 2016; Goodall, 2016; Olson, 2011), it is becoming increasingly important to ensure that trained models are able to avoid bias and discrimination. Observations that the use of ML algorithms might have unexpected social implications, such as bias with respect to sex or race, have led to the creation of algorithms that provide high-confidence fairness guarantees (Thomas et al., 2019; Agarwal et al., 2018). These guarantees rely on the assumption that the data the model is trained with and the data encountered after deployment follow the same distribution. However, this assumption is false for many real-world problems (Zhuang et al., 2021), and, as we demonstrate, models often violate these guarantees and exhibit unfair bias when evaluated on data from a different distribution.

As an example, consider a model that uses university entrance exam scores to predict subsequent success. Because student demographics, such as race or sex, can shift over time, the distribution of applicants may change between model training and deployment. As a result, even if the model is trained to protect a disadvantaged group, if the learning algorithm assumes the training and deployment distributions are the same, many fairness guarantees will not hold in practice (Section 4 empirically verifies this claim). We refer to such distribution change as *demographic shift*. Specifically, demographic shift occurs when the difference between the training and deployment distributions can be explained by a shift in the marginal distribution of a single random variable, such as race or sex.

We present `Shifty`, the first strategy for designing ML algorithms that provide high-confidence guarantees that one or more user-specified fairness constraints will hold despite a demographic shift between training and deployment. We design `Shifty` algorithms to work in two scenarios based on what the user knows at training regarding the demographic shift: **(1)** the old and new demographic proportions are known, or **(2)** the demographic proportions are bounded in known intervals.

We evaluate `Shifty` on a real-world dataset of students' college entrance exam scores and their subsequent grade point average (GPA) in the Brazilian university system (da Silva, 2019). We compare `Shifty` to three families of existing algorithms: Fairness Constraints (Zafar et al., 2017), which aim

to enforce a specific definition of fairness without guarantees, and Seldonian algorithms (Thomas et al., 2019) and Fairlearn (Agarwal et al., 2018), both of which provide high-confidence fairness guarantees.

`Shifty` allows the user to provide (one or multiple simultaneous) fairness definitions from a large class appropriate to the application domain, such as demographic parity, disparate impact, equalized odds, predictive equality, and individual fairness. Unlike all prior algorithms, `Shifty` algorithms provide high-confidence guarantees that the learned model satisfies these fairness constraints even when deployed on a distribution different from the training one. We demonstrate empirically that models learned by previous algorithms do, at times, violate the desired properties under demographic shift, while `Shifty`'s models do not. If there is insufficient training data, or if the specified fairness properties are simultaneously unsatisfiable, `Shifty` algorithms are designed to return no solution, but we show that this rarely happens in practice. Finally and crucially, we demonstrate empirically that, in our evaluated domain, `Shifty` is able to learn models that exhibit no loss of accuracy compared to the models that do not guarantee fairness, as long as sufficient training data exists.

Our main contributions are: **(1)** the first classification algorithms that provide high-confidence fairness guarantees under demographic shift, **(2)** a constructive proof of the guarantees, and a method for creating such algorithms, **(3)** an evaluation on real-world data, and **(4)** an open-source `Shifty` implementation and a release of all our data.

2 Background and Related Work

We illustrate our approach on fair classification, although the methods we propose are easily extended to other problems, such as regression (Thomas et al., 2019) and contextual bandits (Metevier et al., 2019). In this setting, a data *instance* consists of a set of *features* and an associated *label*. When considering the fairness of a classifier, each instance can be augmented with a *fairness attribute*. This information is often not used for prediction (e.g., some laws prohibit the use of race or sex in hiring decisions), but is assumed available for determining if the classifier exhibits bias.

We denote features by $X \in \mathcal{X}$, labels by $Y \in \mathcal{Y}$, and the fairness attribute by $S \in \mathcal{S}$. We assume that (X, Y, S) is sampled from some joint probability distribution defined over $\mathcal{X} \times \mathcal{Y} \times \mathcal{S}$. The naïve classification setting ignores the fairness attribute, and the goal is to accurately predict the label associated with X when its true label is unknown. These predictions are generated using a *model*, $\theta : \mathcal{X} \to \mathcal{Y}$. A loss function, such as expected classification error, measures the quality of θ. To obtain an accurate classifier, one typically selects a training algorithm, a, designed to minimize the chosen loss, and supplies it with a dataset consisting of n observations sampled independently from the joint distribution—that is, $D = \{(X_i, Y_i, S_i)\}_{i=1}^{n}$, where $\Pr(X_i, Y_i, S_i) \coloneqq \Pr(X, Y, S)$ for all $i \in \{1, ..., n\}$.

To assess the fairness of an algorithm, the user provides a function, g, which accepts a model and is calibrated so $g(\theta) > 0$ if and only if θ behaves unfairly. Typically, g depends on the fairness attribute, S. For example, to assess if a classifier is biased based on race, g might measure the difference in the classifier's accuracy for individuals of one race compared to another.

Importantly, g can be defined based on the particular fairness requirements of a given application. Here, we consider the illustrative case where $g(\theta)$ is based on conditional expected value. (Appendix D extends our approach to handle more general definitions of g.) Concretely, let $H \coloneqq h(X, Y, S, \theta)$ define some choice of real-valued observable, let $\xi \coloneqq c(X, Y, S, \theta)$ be some event, and let τ represent a real-valued tolerance. We then assume that g is defined by $g(\theta) \coloneqq \mathbf{E}\big[\, H \,\big|\, \xi \,\big] - \tau$. For example, if for a binary classification problem, the fairness attribute is sex, and the user wants to ensure that the false-positive rate of the model is below 20% for females, one might set $g(\theta) = \mathbf{E}\big[\, \theta(X) \,\big|Y{=}0, S{=}\texttt{female}\big] - 0.2$. While this form of g is not flexible enough to represent many widely used notions of fairness, it serves to illustrate our strategies for accounting for demographic shift, which are straightforward to apply to more sophisticated definitions, as Appendix D shows.

Given a definition for g, we say that a training algorithm, a, is fair with high confidence if

$$\Pr\big(\, g(a(D)) \leq 0 \,\big) \geq 1{-}\delta, \tag{1}$$

for some confidence threshold, $\delta \in [0, 1]$. Allowing the user to set δ during training overcomes the problem that guaranteeing fairness with absolute certainty is often impossible (Thomas et al., 2019).

It is possible that for some g provided by the user, there is no model θ that satisfies $g(\theta) \leq 0$ and, consequently, no algorithm a that satisfies (1). To address this, we adopt the convention described by Thomas et al. (2019) and permit algorithms to return NO_SOLUTION_FOUND (NSF), which is assumed to be fair by definition—that is, we assume that $g(\texttt{NSF}) = 0$. Intuitively, if a fair predictive model cannot be found, such algorithms are permitted to abstain from outputting an unfair predictive model, instead alerting the user that the fairness constraints could not be enforced with the required probability using the data provided. Since the trivial algorithm, $a(D) = \texttt{NSF}$ satisfies (1) by definition, we seek algorithms that satisfy (1) but return useful predictive models as frequently as possible, and we explicitly evaluate this consideration in our experimental designs.

Fair Classification under Demographic Shift: To reason about differences between the training and deployment data distributions, we augment each data instance with a random variable representing a *demographic attribute*, denoted by $T \in \mathcal{T}$. The demographic attribute is often distinct from the other variables defining each observation, but does not need to be. For example, a user might seek a model that avoids unfair bias against individuals of a particular sex, even if the distribution of the individuals' race differs from the training distribution. In this case, S would represent the sex of an individual, while T would represent their race.

Given the demographic attribute, we let (X, Y, S, T) represent an instance observed during training and let (X', Y', S', T') represent an instance encountered once the model is deployed. To formalize the effect of demographic shift, we assume that the demographic attribute's marginal distribution may change between training and deployment, but that the pre- and post-shift joint distributions over the instances are otherwise identical. This can be summarized by the following two conditions, which we refer to as the *demographic shift assumptions*:

$$\exists\, t \in \mathcal{T} \quad \text{s.t.} \quad \Pr(T = t) \neq \Pr(T' = t), \quad \text{and} \tag{2}$$

$$\forall\, (x, y, s, t), \quad \Pr(X{=}x, Y{=}y, S{=}s \,|\, T{=}t) = \Pr(X'{=}x, Y'{=}y, S'{=}s \,|\, T'{=}t). \tag{3}$$

Because the fairness of a model—that is, the value of $g(\theta)$—implicitly depends on X, Y, and S, it follows that guarantees of fairness based on g may fail to hold after the model is deployed, which corresponds to replacing these random variables with X', Y' and S'. Formally, if $H' = h(X', Y', S', \theta)$ and $\xi' = c(X', Y', S', \theta)$, so that $g'(\theta) = \mathbf{E}\big[H' \big| \xi'\big] - \tau$ measures the prevalence of unfair behavior after θ is deployed, then the challenge presented by demographic shift is summarized by the observation that, for all training algorithms a,

$$\underbrace{\Pr\big(\, g(a(D)) \leq 0\big) \geq 1{-}\delta}_{\text{Property A}} \quad \not\Longrightarrow \quad \underbrace{\Pr\big(\, g'(a(D)) \leq 0\big) \geq 1{-}\delta}_{\text{Property B}}.$$

Therefore, we address the following problem:

Problem Statement: Given a user's description of possible demographic shift that might be present between the training and deployment environments as well as one or more definitions of fairness, design an algorithm, a, that provides high-confidence fairness guarantees that a returned model will behave fairly once the model is deployed—that is, an algorithm that satisfies Property B.

In this paper, we assume the user's description of the demographic shift is defined by a set of upper and lower bounds on the marginal probability of each value of the demographic attribute after deployment. Specifically, the user provides an input, $\mathcal{Q} := \{(a_t, b_t)\}_{t \in \mathcal{T}}$, encoding the assumption that $\Pr(T'{=}t) \in [a_t, b_t]$ for all $t \in \mathcal{T}$. However, the approach used by Shifty is general and can be applied for other descriptions of demographic shift as well. Given $\mathcal{Q}$ of this form, we identify two settings in which Shifty can be applied. In the case of *known demographic shift*, which occurs when the user knows the exact post-deployment distribution over the demographic attribute, the user sets $a_t = b_t = \Pr(T'{=}t)$ for each $t \in \mathcal{T}$, and Shifty will be guaranteed not to output unfair models under the specified deployment distribution. In the second setting, the user does not know the exact demographic shift that will occur and instead specifies non-singleton intervals for each $\Pr(T'{=}t)$. We refer to this as the *unknown demographic shift* setting.

Related Work: While many strategies promote fair outcomes in ML models, most existing approaches do not offer fairness guarantees, and none provide guarantees under demographic shift. Appendix A discusses these approaches in detail. At the high level, they include methods based on enforcing hard constraints (Irani, 2015), soft constraints (Zafar et al., 2017; Smits and Kotanchek, 2005), chance-constrained programming techniques (Charnes and Cooper, 1959; Miller and Wagner,

1965; Prékopa, 1970), or by pre- or post-processing training dataset and model predictions (Verma et al., 2021; Salimi et al., 2019; Awasthi et al., 2020). Meanwhile approaches that have considered fairness under distribution shift do not provide guarantees (Lipton et al., 2018).

The closest approach to ours is Seldonian algorithms (Thomas et al., 2019), which allow the user to specify the fairness definition and provide high-probability fairness guarantees—that is, they satisfy Property A. Seldonian algorithms perform well in real-world applications given sufficient data (Thomas et al., 2019; Metevier et al., 2019), but their guarantees, unlike `Shifty`'s, become invalid under demographic shift, as our experiments will show.

3 METHODOLOGY

An overview of `Shifty` is shown in Figure 1. Motivated by the design principles of previous Seldonian algorithms (Thomas et al., 2019), which are effective at designing algorithms that satisfy Property A, `Shifty` algorithms consist of three core components: *data partitioning*, *candidate selection*, and a *fairness test*. First, the data partitioning step splits the input dataset into two parts, which are used to perform the candidate selection and fairness test steps, respectively (see Appendix B). Next, a candidate model is trained. In practice, this can be performed using any existing classification algorithm, as the candidate selection step is not responsible for establishing the fairness guarantees `Shifty` provides. Instead, once a candidate model is found, `Shifty` performs a fairness test by computing a high-confidence upper bound on the prevalence of unfair behavior when the candidate model is deployed in an environment affected by the demographic shift described by $\mathcal{Q}$. If this high-confidence upper bound is below zero, the candidate model is likely to behave fairly once deployed, and the candidate is returned. However, if the value of the high-confidence upper bound is greater than zero, then `Shifty` cannot show that the candidate model will behave fairly with the required confidence, and returns `NSF` instead. Because `Shifty` only returns candidate models if they can be shown to be fair with high confidence on the demographic-shifted deployment distribution, it is guaranteed to satisfy Property B.

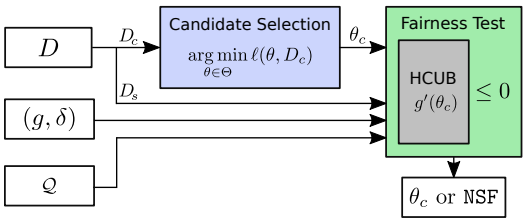


Figure 1: `Shifty` accepts a training dataset, D, one or more fairness specifications (each consisting of a definition of unfair behavior, g, and a tolerance, $\delta \in [0, 1]$), and a description of the possible demographic shifts, $\mathcal{Q}$, that might occur after deployment. It first partitions the input data into D_c and D_f, and uses D_c to select a candidate model, θ_c. Then, it uses $\mathcal{Q}$ and D_f to compute a $(1-\delta)$-probability high-confidence upper bound (HCUB) on the value of $g(\theta_c)$ after deployment, for each fairness definition. If these upper bounds are below zero, `Shifty` returns θ_c, and otherwise returns `NO_SOLUTION_FOUND` (`NSF`). Consequently, `Shifty` returns models that are unfair after deployment with probability at most δ.

Algorithm 1 `Shifty` $(D, g, \delta, \mathcal{Q})$

1: $D_c, D_f \leftarrow$ `Partition`(D)
2: $\theta_c \leftarrow$ `TrainCandidate`$(D_c, g, \delta, \mathcal{Q})$
3: $u \leftarrow$ `HighConfUB`$(\theta_c, g, D_f, \delta, \mathcal{Q})$
4: **return** θ_c **if** $u \leq 0$ **else** `NSF`

Algorithm 1 presents high-level pseudocode for classification algorithms that provide high-confidence fairness guarantees under demographic shift. `TrainCandidate` implements the candidate selection step, and `HighConfUB` implements the high-confidence upper bound on $g'(\theta_c)$ that is used to determine the output of the algorithm and to establish its theoretical guarantees. In the following sections, we describe the candidate selection and fairness test steps in detail. Because it is the fairness test that causes `Shifty` to satisfy Property B, we outline this component first.

3.1 THE SHIFTY FAIRNESS TEST

Given a candidate model, θ_c, `Shifty`'s fairness test consists of computing a $(1-\delta)$-confidence upper bound on $g'(\theta_c)$, which measures the prevalence of unfair behavior once θ_c is deployed. Therefore, the primary challenge associated with designing an algorithm that satisfies Property B is computing a

valid high-confidence upper bound on $g'(\theta_c)$ given $\mathcal{Q}$, the description of the possible demographic distributions that might be encountered upon deployment.

Here, we propose a strategy for computing high-confidence upper bounds on $g'(\theta_c)$, starting with the simpler case in which the exact demographic shift is known and then extending this approach to the case in which the shift is unknown. In both settings, we show how to compute these bounds using appropriate confidence intervals. Our strategy is general and can be applied starting with many different confidence intervals. For illustrative purposes, we derive bounds based on inversion of the Student's t-test (Student, 1908), leading to an implementation of `Shifty` which we call `Shifty-ttest`. Importantly, our use of the Student's t-test implies that the resulting high-confidence upper bounds only hold exactly if $\Pr(H'|\xi)$ is a normal distribution.

3.1.1 KNOWN DEMOGRAPHIC SHIFT

To begin, we consider the task of assessing unfair behavior when there is no demographic shift. Given n i.i.d. samples $\{(H_i, \xi_i)\}_{i=1}^n$, where $H_i \coloneqq h(X_i, Y_i, S_i, \theta)$ and $\xi_i \coloneqq c(X_i, Y_i, S_i, \theta)$, one can derive $U_{\texttt{ttest}}$, a function that computes a high-confidence upper bound on $\mathbf{E}[H|\xi]$, by inverting the commonly-used Student's t-Test (Student, 1908):

$$\Pr\left(\mathbf{E}[H|\xi] \leq U_{\texttt{ttest}}(g, D, \theta, \delta)\right) \geq 1-\delta. \tag{4}$$

Specifically, if $\mathcal{I}_\xi$ are the indices of the samples for which $\xi_i = \texttt{True}$ and $N_\xi = |\mathcal{I}_\xi|$, then

$$U_{\texttt{ttest}}(g, D, \theta, \delta) \coloneqq \frac{1}{N_\xi}\sum_{i\in\mathcal{I}_\xi} H_i + \frac{\sigma(g, D, \theta)}{\sqrt{N_\xi}} t_{1-\delta, N_\xi - 1},$$

where $t_{1-\delta, N_\xi-1}$ is the $1-\delta$ quantile of the Student's t distribution with $N_\xi - 1$ degrees of freedom, and σ computes the sample standard deviation using Bessel's correction,

$$\sigma(g, D, \theta) \coloneqq \sqrt{\frac{1}{N_\xi - 1}\sum_{i\in\mathcal{I}_\xi}\left(H_i - \frac{1}{N_\xi}\sum_{i\in\mathcal{I}_\xi} H_i\right)^2}.$$

As $g(\theta) = \mathbf{E}[H|\xi] - \tau$, it follows that in the absence of demographic shift, Property A is satisfied if algorithm a only returns models satisfying $U_{\texttt{ttest}}(g, D, \theta, \delta) - \tau \leq 0$, and otherwise returns `NSF`.

To provide fairness guarantees that hold under demographic shift, we require $g'(\theta) \leq 0$ with high probability, where $g'(\theta) = \mathbf{E}[H'|\xi'] - \tau$. However, H' and ξ' are defined with respect to the demographic-shifted distribution, for which no samples are available. Thus, we seek a new random variable, $\hat{H}$, that can be computed from X, Y, and S, but which satisfies $\mathbf{E}[\hat{H}|\xi] = \mathbf{E}[H'|\xi']$. Under the demographic shift assumptions, the reweighted variable defined by $\hat{H} \coloneqq \phi(T)H$ satisfies these requirements, where ϕ is an *importance weight* derived in Appendix C:

$$\phi(t) \coloneqq \frac{\Pr(T'{=}t|\xi')}{\Pr(T{=}t|\xi)} = \frac{\Pr(\xi|T{=}t)\Pr(T'{=}t)}{\Pr(T{=}t|\xi)\sum_{t'\in\mathcal{T}}\Pr(\xi|T{=}t')\Pr(T'{=}t')}, \tag{5}$$

for all $t \in \mathcal{T}$. Specifically, $\phi(T)$ acts as a scaling factor that reweights samples obtained during training so that their sample mean is an unbiased estimator of $\mathbf{E}[H'|\xi']$, as shown by Theorem 1.

Theorem 1. *Assume that* $\Pr(T{=}t) \geq 0$ *for all* $t \in \mathcal{T}$. *If the demographic shift properties hold, then the random variable* $\hat{H} \coloneqq \phi(T)H$ *satisfies* $\mathbf{E}[\hat{H}|\xi] = \mathbf{E}[H'|\xi']$, *where* ϕ *is defined by (5).* ***Proof.*** *See Appendix C.*

Because $\hat{H}$ is defined with respect to pre-shift random variables, it is possible to generate i.i.d. samples of $\hat{H}$ during training, even when no samples from the deployment distribution are available. In particular, a set of i.i.d. observations of $\hat{H}$ is obtained by computing $\{\hat{H}_i\}_{i\in\mathcal{I}_\xi}$, where each $\hat{H}_i = \phi(T_i)h(X_i, Y_i, S_i, \theta)$. Using $\{\hat{H}_i\}_{i\in\mathcal{I}_\xi}$, we apply the inversion of the Student's t-test to derive $\hat{U}_{\texttt{ttest}}(g, D, \theta, \delta)$, which satisfies $\Pr(\mathbf{E}[\hat{H}|\xi] \leq \hat{U}_{\texttt{ttest}}(g, D, \theta, \delta)) \geq 1-\delta$. Specifically, if $\hat{\sigma}$ denotes the sample standard deviation of the reweighted observations, $\{\hat{H}_i\}_{i\in\mathcal{I}_\xi}$, then

$$\hat{U}_{\texttt{ttest}}(g, D, \theta, \delta) \coloneqq \frac{1}{N_\xi}\sum_{i\in\mathcal{I}_\xi}\hat{H}_i + \frac{\hat{\sigma}(g, D, \theta)}{\sqrt{N_\xi}} t_{1-\delta, N_\xi-1}. \tag{6}$$

Since $\mathbf{E}[\hat{H}|\xi] = \mathbf{E}[H'|\xi']$ by Theorem 1, it follows that $\hat{U}_{\texttt{ttest}}$ is also a high-confidence upper bound suitable for assessing fairness after demographic shift:

$$\Pr\left(\mathbf{E}[\hat{H}|\xi] \leq \hat{U}_{\texttt{ttest}}(g, D, \theta, \delta)\right) = \Pr\left(\mathbf{E}[H'|\xi'] \leq \hat{U}_{\texttt{ttest}}(g, D, \theta, \delta)\right) \geq 1{-}\delta. \tag{7}$$

Recalling that $g'(\theta) \coloneqq \mathbf{E}[H'|\xi'] - \tau$, it follows that $g'(\theta) \leq 0$ with high probability if $\hat{U}_{\texttt{ttest}}(g, D, \theta, \delta) - \tau \leq 0$, where each $\hat{H}_i$ implicitly depends on θ by the definition, $\hat{H}_i \coloneqq \phi(T_i)h(X_i, Y_i, S_i, \theta)$. From (7), it is clear that if the pre-shift conditionals, $\Pr(\xi|T{=}t)$ and $\Pr(T{=}t|\xi)$, can be computed from the training data for all $t \in \mathcal{T}$, and if the post-shift demographic marginals, $\Pr(T'{=}t)$, are provided by the user during training, then a $(1{-}\delta)$-confidence upper bound $g'(\theta)$ can be computed even when data from the post-shift distribution is unavailable.

3.1.2 Unknown Demographic Shift

It is often unrealistic to assume that the post-shift marginal distribution is known exactly during training. To address this, we consider the setting in which the user provides a set, $\mathcal{Q} \coloneqq \{(a_t, b_t)\}_{t\in\mathcal{T}}$, that contains non-empty intervals describing marginal distributions over T' that might be encountered after deployment. Given $\mathcal{Q}$, we compute high-confidence upper bounds on $g'(\theta)$ by determining the largest value of the high-confidence upper bound attained for a $q \in \mathcal{Q}$.

First, we parameterize the high-confidence upper bound in (7) to explicitly depend on a particular choice of post-shift demographic distribution, q. Formally, we define $\hat{U}_{\texttt{ttest}}(g, D, \theta, \delta; q)$ by replacing all occurrences of $\phi(T_i)$ in (6) with $\phi(T_i; q)$, given by

$$\phi(t; q) \coloneqq \frac{\Pr(\xi|T{=}t)q_t}{\Pr(T{=}t|\xi)\sum_{t'\in\mathcal{T}}\Pr(\xi|T{=}t')q_{t'}}, \tag{8}$$

where $q_t = \Pr(T'{=}t)$ for one possible demographic-shifted demographic distribution. While the true post-shift marginal distribution, q^*, is assumed to be unknown, it is clear that if $q^* \in \mathcal{Q}$, then $\hat{U}_{\texttt{ttest}}(g, D, \theta, \delta; q^*) \leq \sup_{q\in\mathcal{Q}} \hat{U}_{\texttt{ttest}}(g, D, \theta, \delta; q)$. It follows that,

$$\Pr\left(\mathbf{E}[H'|\xi'] \leq \sup_{q\in\mathcal{Q}} \hat{U}_{\texttt{ttest}}(g, D, \theta, \delta; q)\right) \geq \Pr\left(\mathbf{E}[H'|\xi'] \leq \hat{U}_{\texttt{ttest}}(g, D, \theta, \delta; q^*)\right) \geq 1{-}\delta.$$

Consequently, if $g'(\theta) \coloneqq \mathbf{E}[H'|\xi'] - \tau$, then an algorithm, a, can be designed to satisfy Property B by following Algorithm 1 and defining the fairness test to only return models when $\sup_{q\in\mathcal{Q}} \hat{U}_{\texttt{ttest}}(g, D, \theta, \delta; q) - \tau \leq 0$. We propose to use a numerical optimizer to approximate the supremum of $\hat{U}_{\texttt{ttest}}$ over $q \in \mathcal{Q}$. In our implementations, we use *simplicial homology optimization* (Endres et al., 2018), which converges to the global optima of non-smooth functions subject to equality and inequality constraints such as those defined by the condition $q \in \mathcal{Q}$.

3.2 Candidate Selection

The candidate selection step searches over a set of candidate models to find a model, θ_c, that achieves a small classification error. Because it is ultimately the fairness test that causes `Shifty` to satisfy Property B, this step can be implemented using any procedure for training a classifier without impacting the theoretical guarantees the algorithm provides. In practice, however, it is advantageous to select a candidate model that, in addition to minimizing error, also appears to behave fairly. In particular, if the model's accuracy is correlated with the definition of unfair behavior, a candidate selection procedure that solely optimizes accuracy will tend to select models that will fail the fairness test, causing the overall algorithm to frequently return `NSF`.

To mitigate this problem, we perform candidate selection by minimizing a loss consisting of two terms: one that measures classification error, and another that penalizes models that are predicted to fail the fairness test. Specifically, if $\mathbb{I}[\cdot]$ denotes the indicator function that returns 1 if its argument is `true` and 0 otherwise, then the candidate model is found by minimizing,

$$\ell_{\texttt{Shifty-ttest}}(g, D_c, \theta, \delta; \mathcal{Q}) \coloneqq \sum_{(x,y)\in D_c} \frac{\mathbb{I}[\theta(x){\neq}y]}{|D_c|} + \max\left(0, \sup_{q\in\mathcal{Q}} \hat{U}_{\texttt{ttest}}(g, D_c, \theta, \delta; q)\right). \tag{9}$$

3.3 IMPLEMENTATION DETAILS FOR SHIFTY-TTEST

Having described strategies for computing high-confidence upper bounds on the prevalence of unfair behavior under demographic shift, we now present the details for `Shifty-ttest`, an implementation `Shifty` based on the Student's t-Test. Specifically, `Shifty-ttest` is designed according to Algorithm 1, where the subroutine `TrainCandidate` minimizes (9), and `HighConfUB` computes $\sup_{q \in \mathcal{Q}} \hat{U}_{\texttt{ttest}}(g, D, \theta, \delta; q) - \tau$. Consequently, `Shifty-ttest` produces binary classifiers that are fair under demographic shift by leveraging user-provided bounds on the marginal distribution of the demographic attribute after deployment.

4 EVALUATION

Our evaluation answers three research questions (RQ) regarding our algorithms' behavior for overcoming demographic shift compared to prior approaches that do not account for demographic shift.

[RQ1] Validity of fairness guarantees. In practice, do the models trained using `Shifty-ttest` or prior approaches adhere to high-probability fairness guarantees under demographic shift? **[RQ2] Model accuracy**. Is `Shifty-ttest` able to train models whose accuracy is comparable to those produced by prior approaches that do not account for demographic shift? **[RQ3] Data efficiency**. Does `Shifty-ttest` avoid returning `NSF` when reasonably sized training datasets are available?

Answering these research questions requires *multiple* pairs of datasets sampled from the same underlying distribution and exhibiting a consistent demographic shift. Therefore, we generate multiple training and deployment datasets by resampling from a fixed population using known distributions. This design ensures that failures—that is, instances in which an algorithm returns unfair models with a larger frequency than guaranteed—can be properly attributed to a failure of the algorithm instead of a violation of the user's assumptions on the demographic shift. In addition, oracle knowledge of the training and deployment distributions can be used to compute exact values for accuracy, $g(\theta)$, and $g'(\theta)$ during evaluation. We uniformly sample training datasets from the population, train models using each algorithm, and evaluate the fairness of each model after deployment by selecting a new distribution over the population that satisfies the user's assumptions about the demographic shift.

Dataset: We sample training and deployment data from a dataset of $43{,}303$ students from a university in Brazil (da Silva, 2019). The dataset consists of tuples (X, Y, S, T), where X is a vector of entrance exam scores, Y is a binary label representing if the student's GPA, between 0 and 4.0, was above 3.0, the demographic attribute, T, is the student's race, and S is the student's sex. This data allows us to train classifiers to predict academic success based on entrance exam scores, and assess whether fairness guarantees that protect against discrimination based on sex would continue to hold if demographic shift caused the marginal distribution over race to change during deployment.

4.1 EXPERIMENTS: KNOWN DEMOGRAPHIC SHIFT

First, we evaluate `Shifty-ttest`, prior Seldonian and Quasi-Seldonian algorithms (Thomas et al., 2019), Fairlearn (Agarwal et al., 2018), and Fairness Constraints (Zafar et al., 2017). We conducted experiments using these algorithms to enforce two fairness definitions—disparate impact and demographic parity. Figure 5 formalizes these definitions, which were selected as representative of real-world strategies for quantifying unfair behavior of classifiers (Verma and Rubin, 2018). To assess the impact of demographic shift, we simulated one possible choice of demographic shift: the left column of Figure 2 shows the uniformly sampled distribution used for training, and the middle column shows the demographic-shifted deployment distribution used in our experiments.

For each experiment, we conducted 15 trials while varying the amount of training data in order to identify dependencies on the training dataset's size. For each trial, we uniformly sampled a training dataset and trained models using our algorithm, standard Seldonian algorithms, Fairlearn, and Fairness Constraints. Importantly, while Fairlearn was not designed to avoid disparate impact, and Fairness Constraints was not designed to enforce either disparate impact or demographic parity constraints, we include them to test if they might empirically be fair under demographic shift, despite not being designed for this setting. After training, we recorded whether each algorithm produced a model or `NSF`, as well as the average accuracy and prevalence of unfair behavior for each trained model on both the training and demographic-shifted deployment distribution.

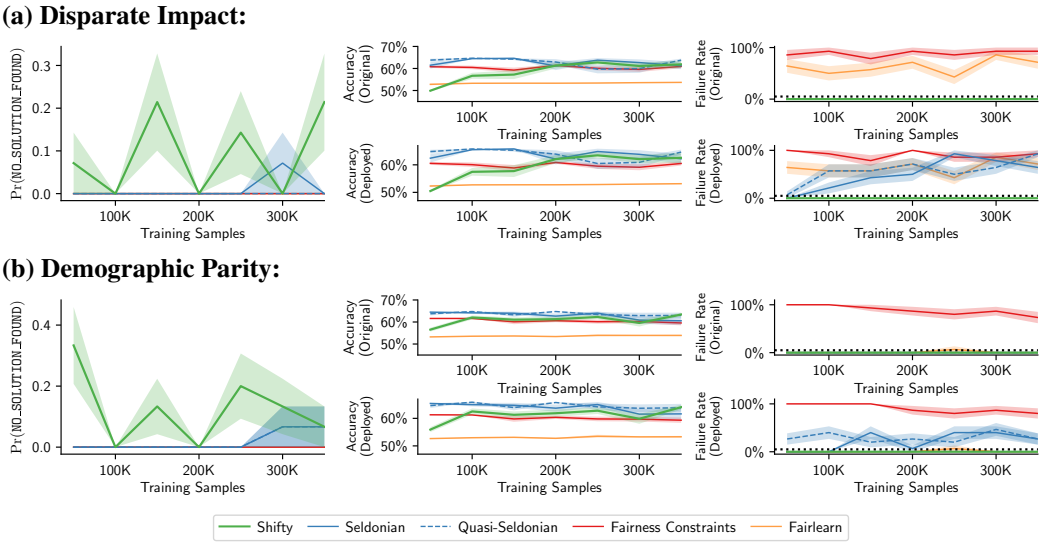


Figure 3: Results when enforcing fairness constraints under known demographic shift.

Figure 3 shows the results. For each definition of fairness, the leftmost plot shows the probability with which each algorithm returns `NSF`. To the right are two rows of plots showing evaluations of each algorithm on the training and deployment distributions, respectively. Within each row, the left plot displays the average accuracy of models returned by each algorithm when trained using various amounts of data. The right plot in each row displays the frequency with which each algorithm returns a model that is unfair, which we call the failure rate. The horizontal dashed line shows the maximum tolerance for unfair outcomes, δ=0.05, set when training `Shifty-ttest` and the Seldonian algorithms. Finally, standard error bars are shown using shaded regions.

Race	$\Pr(T{=}t)$	$\Pr(T'{=}t)$	$\mathcal{Q}$
$race_1$	0.006	0.300	(0.004, 0.304)
$race_2$	0.871	0.600	(0.610, 0.909)
$race_3$	0.054	0.050	(0.038, 0.338)
$race_4$	0.067	0.048	(0.047, 0.347)
$race_5$	0.002	0.002	(0.002, 0.301)

Figure 2: Marginal distributions over anonymized student race. The left column shows the distribution during training. The middle column shows the deployed marginal distribution used in our experiments for known demographic shift (Section 4.1). The right column shows the bounds on the marginal distribution used in our experiments for unknown demographic shift (Section 4.2).

Our experiments confirm that `Shifty` algrithms effectively avoid unfair behavior after demographic shift while prior algorithms do not. The failure rates of `Shifty-ttest` after demographic shift occurs (bottom right plots in Figure 3) are **always** below the tolerance set during training. However, while standard Seldonian algorithms were fair during training, they, along with Fairlearn and Fairness Constraints, frequently violate that fairness constraint after deployment. Interestingly, Fairlearn models were also fair when enforcing demographic parity constraints, though this was not true for disparate impact. However, Fairlearn does not provide fairness guarantees, unlike `Shifty-ttest`.

Next, when provided with a sufficient amount of training observations, `Shifty-ttest` provided guarantees of fairness without a noticeable loss in accuracy compared to the other baselines. The plots in the middle column of Figure 3 show that `Shifty-ttest` produced models that achieved the same accuracy as those trained using standard Seldonian algorithms, and outperformed both Fairlearn and Fairness Constraints for sufficiently large training datasets. This provides evidence that `Shifty-ttest` can be used to train fair models without incurring a significant loss in accuracy compared to algorithms that do not provide such guarantees.

Finally, we found that `Shifty-ttest`'s data efficiency is lower than that of alternative algorithms that ignore the impact of demographic shift. `Shifty-ttest` required slightly more training data than standard Seldonian algorithms to consistently avoid returning `NSF` (left plots in Figure 3) and to achieve comparable accuracy to those methods.

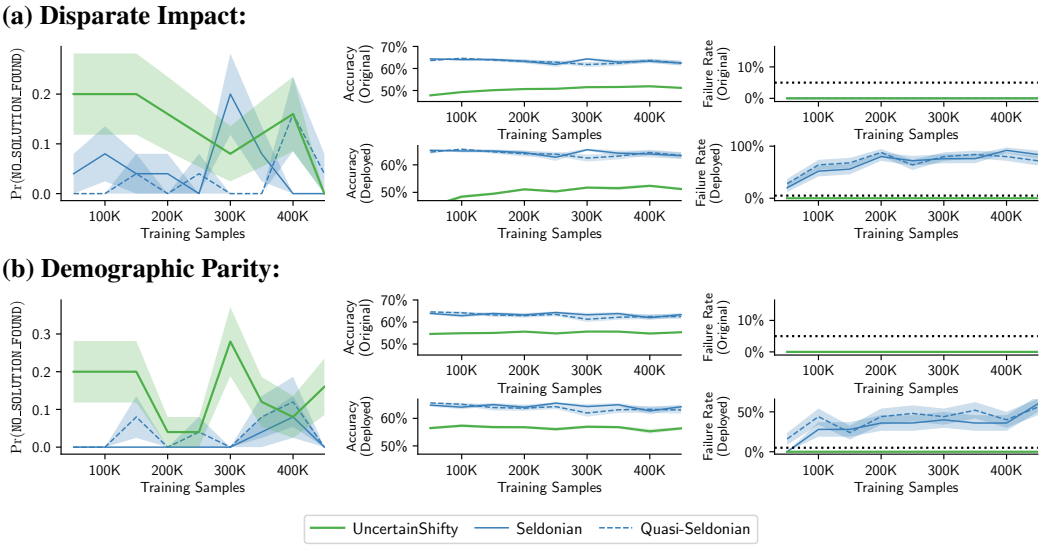


Figure 4: Results when enforcing fairness constraints under unknown demographic shift.

4.2 Experiments: Unknown Demographic Shift

Next, we repeat our experiments from Section 4.1, but assume that the user has provided upper and lower bounds on the marginal probability of encountering individuals of each race after deployment instead of specifying the shift exactly. We generated these bounds, shown in the rightmost column of Figure 2, by interpolating between the interval $(0, 1)$ and the singleton interval containing the marginal distribution over race under the training distribution, using an interpolation factor of 0.3. Given this specification, we applied each algorithm to train models using the same procedure as described in Section 4.1. To evaluate each model's performance after demographic shift, we performed a worst-case analysis and selected the deployment distribution over the population to satisfy the user's assumptions, but otherwise *maximize* the prevalence of unfair behavior when using that model to generate predictions (See Appendix E). Figure 4 in Appendix 6 shows the results for disparate impact and demographic parity, and Appendix F provides results based on three other definitions of unfair behavior. As in Section 4.1, we include five plots that show the probability of each algorithm returning `NSF`, average accuracy, and the failure rate of each algorithm before and after deployment.

These experiments confirm that `Shifty-ttest` can provide guarantees of fair behavior after deployment, even when the post-shift marginal distribution over race is not exactly known. While standard Seldonian algorithms are fair during training (Figure 4, top right), they are consistently unfair after demographic shift occurs. However, `Shifty-ttest` was unable to produce models that achieved the same accuracy as standard Seldonian algorithms. Some loss in accuracy is expected when enforcing constraints that hold under demographic shift, which may be more limiting than standard fairness constraints. Therefore, `Shifty-ttest` is effective when fairness guarantees are critical, but these guarantees may cause decreased classification accuracy in some settings.

5 Conclusion

In this paper, we proposed `Shifty`, a strategy for designing classification algorithms that provide high-confidence fairness guarantees that hold when the distribution of demographics changes between training and deployment. This setting poses significant challenges for existing fair ML algorithms, as the fairness guarantees they provide generally assume a constant data distribution. In contrast, `Shifty` algorithms allow the proportions of demographics to change after training, provided the user has some information describing this change. `Shifty` algorithms can be used when the new demographic proportions are known, or when these proportions are bounded in known intervals. Finally, we evaluated `Shifty-ttest`, an implementation of `Shifty` based on the Student's t-test, and found that the fairness guarantees it provides are empirically valid under demographic shift, wherein models trained using existing fair algorithms consistently produced unfair outcomes.

ETHICS STATEMENT

The primary goal of this research is to identify and overcome practical challenges, namely demographic shift, that might cause current algorithms to produce unfair outcomes. Our contributions provide tools needed for data scientists and ML practitioners to use ML in conscientious, ethical ways. However, we note that, when applying algorithms such as `Shifty`, it is important to carefully select the definition of unfair behavior to be appropriate for the problem at hand. While we evaluate `Shifty` using five standard definitions of unfair behavior for illustration, many definitions have been proposed and studied (Verma and Rubin, 2018), some of which cannot be simultaneously enforced (Chouldechova, 2017; Corbett-Davies et al., 2017; Kleinberg et al., 2017). Consequently, while `Shifty` offers a valuable tool for enforcing fairness constraints, ML practitioners should carefully study their target application, ideally working with domain experts and stakeholders, to ensure that the definitions they select meaningfully capture the unfair behaviors they wish to avoid.

Next, we note that the fairness guarantees provided by `Shifty` may fail to hold if one or more assumptions made by the algorithm do not hold. Most notably, if the demographic shift assumptions, (2) and (3), do not hold for the user's choice of demographic attribute T, then the guarantees provided by `Shifty` may not hold in practice. Similarly, if the user's specification of possible demographic shift, defined by the input $\mathcal{Q}$, does not accurately represent the demographic shift that occurs, then the guarantees provided by our algorithms may be invalidated. Thus, it is important that ML practitioners study the application domain in order to ensure that the inputs they supply to `Shifty` are appropriate.

Finally, `Shifty-ttest` is based on inversion of the Student's t-test, which only holds exactly if the observations, H_i are not normal. While this approximation error becomes smaller as more samples are used for training, the high-confidence guarantees provided by `Shifty` may not hold with $1-\delta$ probability when trained on very small datasets. Therefore, in applications for which very few training observations are available, ML practitioners should employ `Shifty` algorithms based on other, non-approximate confidence intervals.

REPRODUCIBILITY STATEMENT

To support efforts to reproduce our results, all code and data used in this paper will be made publicly available upon publication. Proofs of our theoretical results can be found in Appendix C and implementation details for our algorithms can be found in Section 3 and Appendix D. In addition, experimental details can be found in Section 4, full descriptions of the fairness definitions we tested are shown in Figure 5 of Appendix F, and additional results and experimental details are included in Appendix E.

REFERENCES

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning (ICML)*, Vol. PMLR 80. Stockholm, Sweden, 60–69.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. *ProPublica* (23 May 2016). `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`

Pranjal Awasthi, Matthäus Kleindessner, and Jamie Morgenstern. 2020. Equalized odds postprocessing under imperfect group information. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Palermo, Italy.

Arpita Biswas and Suvam Mukherjee. 2021. Ensuring fairness under prior probability shifts. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 414–424.

Lorenzo Bruzzone and Mattia Marconcini. 2009. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 5 (2009), 770–787.

Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21, 2 (2010), 277–292.

A. Charnes and W. W. Cooper. 1959. Chance-constrained programming. *Management Science* 6, 1 (1959), 73–79.

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

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *ACM Conference on Knowledge Discovery and Data Mining (KDD)*. 797–806.

Amanda Coston, Karthikeyan Natesan Ramamurthy, Dennis Wei, Kush R Varshney, Skyler Speakman, Zairah Mustahsan, and Supriyo Chakraborty. 2019. Fair transfer learning with missing protected attributes. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 91–98.

Bruno Castro da Silva. 2019. UFRGS Entrance Exam and GPA Data. https://doi.org/10.7910/DVN/O35FW8

Jessica Dai. 2020. Label Bias, Label Shift: Fair Machine Learning with Unreliable Labels.

Wei Du and Xintao Wu. 2021. Fair and Robust Classification Under Sample Selection Bias. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2999–3003.

Miroslav Dudík, Steven J Phillips, and Robert E Schapire. 2006. Correcting sample selection bias in maximum entropy density estimation. In *Advances in Neural Information Processing Systems*. 323–330.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*. ACM, 214–226.

Stefan C Endres, Carl Sandrock, and Walter W Focke. 2018. A simplicial homology algorithm for Lipschitz optimisation. *Journal of Global Optimization* 72, 2 (2018), 181–217.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. 2014. Subspace alignment for domain adaptation. *arXiv preprint arXiv:1409.5241* (2014).

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2066–2073.

Noah J. Goodall. 2016. Can you program ethics into a self-driving car? *IEEE Spectrum* 53, 6 (June 2016), 28–58. https://doi.org/10.1109/MSPEC.2016.7473149

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2011. Domain adaptation for object recognition: An unsupervised approach. In *International Conference on Computer Vision*. IEEE, 999–1006.

Griggs v. Duke Power Co. 1971. 401 U.S. 424. https://supreme.justia.com/cases/federal/us/401/424/.

M. Hardt, E. Price, and N. Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Conference on Neural Information Processing Systems (NIPS)*. Barcelona, Spain, 3323–3331.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. 2006. Correcting sample selection bias by unlabeled data. *Advances in Neural Information Processing Systems* 19 (2006), 601–608.

Arja John Irani. 2015. *Utilizing Negative Policy Information to Accelerate Reinforcement Learning*. Ph.D. Dissertation. Georgia Institute of Technology.

Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *Innovations in Theoretical Computer Science Conference (ITCS)*, Vol. 67. Berkeley, CA, USA, 43:1–43:23. https://doi.org/10.4230/LIPIcs.ITCS.2017.43

Ronny Kohavi and Barry Becker. 1996. Uci machine learning repository: adult data set. *Avaliable: https://archive. ics. uci. edu/ml/machine-learning-databases/adult* (1996).

Zachary C Lipton, Yu-Xiang Wang, and Alex Smola. 2018. Detecting and correcting for label shift with black box predictors. *arXiv preprint arXiv:1802.03916* (2018).

Blossom Metevier, Stephen Giguere, Sarah Brockman, Ari Kobren, Yuriy Brun, Emma Brunskill, and Philip S Thomas. 2019. Offline Contextual Bandits with High Probability Fairness Guarantees. In *Advances in Neural Information Processing Systems*. 14922–14933.

L. B. Miller and H. Wagner. 1965. Chance-constrained programming with joint constraints. *Operation Research* 13 (1965), 930–945.

Parmy Olson. 2011. The Algorithm That Beats Your Bank Manager. *CNN Money* March 15 (2011).

A. Prékopa. 1970. On probabilistic constrained programming. In *Princeton Symposium on Mathematical Programming*. 113–138.

Ashkan Rezaei, Anqi Liu, Omid Memarrast, and Brian D. Ziebart. 2021. Robust Fairness Under Covariate Shift. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 11 (May 2021), 9419–9427. https://ojs.aaai.org/index.php/AAAI/article/view/17135

Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional Fairness: Causal Database Repair for Algorithmic Fairness. In *International Conference on Management of Data (SIGMOD)*. Association for Computing Machinery, Amsterdam, Netherlands, 793–810. https://doi.org/10.1145/3299869.3319901

Candice Schumann, Xuezhi Wang, Alex Beutel, Jilin Chen, Hai Qian, and Ed H Chi. 2019. Transfer of machine learning fairness across domains. *arXiv preprint arXiv:1906.09688* (2019).

Harvineet Singh, Rina Singh, Vishwali Mhasawade, and Rumi Chunara. 2021. Fairness violations and mitigation under covariate shift. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 3–13.

Guido F Smits and Mark Kotanchek. 2005. Pareto-front exploitation in symbolic regression. In *Genetic Programming Theory and Practice II*. Springer, 283–299.

Student. 1908. The probable error of a mean. *Biometrika* (1908), 1–25.

Philip S. Thomas, Bruno Castro da Silva, Andrew G. Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. 2019. Preventing undesirable behavior of intelligent machines. *Science* 366, 6468 (2019), 999–1004.

Sahil Verma, Michael Ernst, and René Just. 2021. Removing biased data to improve fairness and accuracy. *CoRR* abs/2102.03054 (2021). https://arxiv.org/abs/2102.03054

Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*. IEEE, 1–7.

Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine learning*. 114.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 797–806.

Xiang Zhang and Yann LeCun. 2017. Universum prescription: Regularization using unlabeled data. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2021. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* 109, 1 (2021), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

Daniel Zwillinger and Stephen Kokoska. 1999. *CRC standard probability and statistics tables and formulae*. CRC Press. 31 pages.

A ALTERNATIVE APPROACHES FOR ACHIEVING FAIR OUTCOMES

Designing algorithms that meet fairness requirements with high-confidence can be seen as a type of constrained optimization problem. Algorithms that enforce *hard constraints* search for a predictive model within a feasible set of models satisfying the user's fairness constraints (Irani, 2015). Soft constraints on the objective function used to guide the search for a model can achieve fairness empirically (Zafar et al., 2017), and multi-objective methods can be used to satisfy multiple, potentially conflicting fairness objectives (Smits and Kotanchek, 2005). Unfortunately, algorithms based on hard or soft constraints alone do not provide high-confidence fairness guarantees. Fair algorithms can be designed based on *chance-constrained programming* (Charnes and Cooper, 1959; Miller and Wagner, 1965; Prékopa, 1970), in which an objective function is optimized subject to a set of constraints. This formulation can provide high-confidence fairness guarantees but typically requires knowledge of the distribution of each variable used to quantify fairness, which is often impractical. In contrast, our methods provide assurances of fair behavior without these limiting assumptions.

In our work, we consider the problem of establishing high-confidence fairness guarantees that hold under demographic shift, specifically when no samples are available from the deployment distribution. While `Shifty` is, to our knowledge, the first to provide high-confidence guarantees of fairness in this setting, there are several existing strategies for promoting fair outcomes under various forms of distribution shift, which leverage a variety of assumptions. For example, approaches have been proposed to promote fair outcomes under *concept shift*, which occurs when the distribution over safety attributes and feature vectors changes between training and deployment. Schumann et al. (2019) derive bounds describing how well fairness properties can be transferred from a source to a target domain, and use them to propose training procedures that can improve the transferability of fairness properties. While this approach could be applied to overcome demographic shift, it does not provide guarantees that the resulting models will meet specific fairness tolerances, and it assumes that data is available from the deployment distribution and therefore cannot be applied in the setting we consider. In contrast to methods for overcoming concept shift, many works on promoting fairness under distribution shift assume that the shift has a particular structure. For example, Dai (2020) propose a fairness-aware model for *label shift* and *label bias*, which occur when the distribution over the fairness attribute and feature vector is the same during training and after deployment, but the conditional distribution over true labels differs. Similarly, Biswas and Mukherjee (2021) propose an algorithm for promoting fair outcomes under *prior probability shift*, which occurs when the marginal distribution of true labels changes between training and deployment, using *proportional equality* to measure fairness. Unfortunately, the assumptions of label shift, label bias, and proportional equality are contrary to the assumptions of demographic shift, so that these approaches cannot be applied to the problem setting we consider.

Among the problem settings considered in prior work, the most similar setting to demographic shift is *covariate shift*, which occurs when the distribution of fairness attributes and feature vectors changes between training and deployment, but the conditional distribution over labels is unchanged. Indeed, if, in addition to the demographic shift assumptions, the user assumes that the distribution of true labels is conditionally independent of the demographic attribute given the features and safety attributes, then demographic shift can be viewed as a form of covariate shift. Several approaches have been proposed for promoting fair outcomes under covariate shift and various definitions of fairness, but these approaches cannot be applied to our problem setting because they make the assumption that samples are available from the deployment distribution. Furthermore, these approaches are not proven to satisfy Property B (Rezaei et al., 2021; Coston et al., 2019). Singh et al. (2021) propose a training algorithm that ensures that fairness properties of the resulting model are invariant to covariate shift by exploiting a causal graph of the problem that is provided by the user. While this approach can be applied without samples from the deployment distribution, the causal graph required by the algorithm is often unknown, and the algorithm does not provide fairness guarantees. Finally, Du and Wu (2021) propose an algorithm, RFLearn, that promotes fair outcomes under covariate shift without access to data from the deployment environment. While this approach does not provide fairness guarantees, and despite the differences between demographic shift and covariate shift, we include RFLearn in our experimental designs for illustration.

We propose algorithms that produce fair models following the design principles outlined by Seldonian ML (Thomas et al., 2019) while accounting for demographic shift. Other ways to account for demographic shift include augmenting training data with synthetically generated variations or

antagonistic examples, or explicitly regularizing the training objective using unlabeled data from the deployment distribution, which improves the models' generalization (Goodfellow et al., 2016; Zhang and LeCun, 2017). However, these approaches do not tackle fairness and cannot be directly applied to provide fairness guarantees under demographic shift. If the change in distribution can be addressed by a transformation of either the features or response variables, methods can learn this transformation using access to data from the deployment distribution combined with assumptions, such as that the transformation is linear (Fernando et al., 2014; Gong et al., 2012; Gopalan et al., 2011), or by iteratively assigning predicted labels to unlabeled data from the deployment environment (Bruzzone and Marconcini, 2009). However, such approaches are ill-suited for establishing high-confidence fairness guarantees under demographic shift.

Approaches can account for the differences between the training and deployment environments by reweighting the contribution of each training observation according to the relative probability of encountering that observation. Such methods are effective at improving performance in classification (Zadrozny, 2004; Huang et al., 2006), density estimation (Dudík et al., 2006), and regression (Huang et al., 2006). While most of these approaches focus on accuracy, some have focused on safety constraints, such as fairness. Lipton et al. (2018) proposed correcting for covariate shift by using a reweighting scheme to compute intervals on a classifier's confusion rates and applying a correction step based on these intervals. This approach produces empirically fair models but does not provide high-confidence guarantees. An algorithm's fairness can be improved by manipulating the underlying data, e.g., by removing data that violates fairness properties (Verma et al., 2021) or inserting data inferred using fairness properties (Salimi et al., 2019). These methods, however, do not provide guarantees. Fairness can also be enforced in post-processing (Awasthi et al., 2020) but, again, without guarantees.

B Data Partitioning

The data partitioning step ensures that the outcome of the candidate selection step is independent of the outcome of the fairness test, which is necessary to guarantee that the overall algorithm satisfies (1) under demographic shift.

To illustrate the requirement for this step, consider an algorithm that accepts a set of training observations, D, uses D to select a candidate model, θ_c, and finally uses D again to evaluate the high-confidence upper bound on $g'(\theta_c)$ to perform a fairness test. While this algorithm might appear to be fair, it is not guaranteed to satisfy Property B because the output of candidate selection, θ_c, is correlated with the outcome of the fairness test because the same input data, D, is used to perform both steps. Consequently, it is possible that candidate selection can consistently select models that cause the high-confidence upper bound used by the fairness test to under-estimate the value of $g'(\theta_c)$. By ensuring that candidate selection and the fairness test use independent sets of observations, this correlation is eliminated, causing the probability of the high-confidence upper bound failing to be no more than δ as required by Property B.

Finally, we note that one area of future research might consider the optimal way to split an input dataset into parts used for candidate selection, D_c, and for evaluating the fairness test, D_f. Specifically, increasing the size of D_c improves the ability of the candidate selection step to identify models that are accurate and generalize well, but reduces the size of D_f and makes the fairness test more difficult to pass. In our experiments, we split the input data evenly between D_c and D_f, but we hypothesize that there may be more effective techniques for determining the optimal splitting proportion.

C Proof of Theorem 1

Theorem 1. *Assume that* $\Pr(T{=}t) \geq 0$ *for all* $t \in \mathcal{T}$. *If the demographic shift properties hold, then the random variable* $\hat{H} \coloneqq \phi(T)H$ *satisfies* $\mathbf{E}[H'|\xi'] = \mathbf{E}[\hat{H}|\xi]$, *where* ϕ *is defined by (5).*

Proof. First, we write $\mathbf{E}[H'|\xi']$ as a sum over expected values conditioned on the value of the demographic attribute by applying the law of total probability (Zwillinger and Kokoska, 1999):

$$\begin{aligned}\mathbf{E}[H'|\xi'] &= \sum_{t\in\mathcal{T}} \mathbf{E}[H'|\xi', T'{=}t] \Pr(T'{=}t|\xi') \\ &= \sum_{t\in\mathcal{T}} \mathbf{E}[H|\xi, T{=}t] \Pr(T'{=}t|\xi'). && \textit{(Using (3))}\end{aligned}$$

Here, the second line follows from the second demographic shift assumption, which states that for all $t \in \mathcal{T}$ and $x, y, s \in \mathcal{X} \times \mathcal{Y} \times \mathcal{S}$, $\Pr(X'{=}x, Y'{=}y, S'{=}s|T'{=}t) = \Pr(X{=}x, Y{=}y, S{=}s|T{=}t)$. Next, we multiply each term by $\Pr(T{=}t|\xi)/\Pr(T{=}t|\xi) = 1$, reorganize terms, and write the sum over $t \in \mathcal{T}$ as a single expected value:

$$\begin{aligned}\mathbf{E}[H'|\xi] &= \sum_{t\in\mathcal{T}} \mathbf{E}[H|\xi, T{=}t] \left(\frac{\Pr(T{=}t|\xi)}{\Pr(T{=}t|\xi)}\right) \Pr(T'{=}t|\xi') \\ &= \sum_{t\in\mathcal{T}} \mathbf{E}[H|\xi, T{=}t] \left(\frac{\Pr(T'{=}t|\xi')}{\Pr(T{=}t|\xi)}\right) \Pr(T{=}t|\xi) \\ &= \mathbf{E}\left[\phi(T)H|\xi\right],\end{aligned}$$

where

$$\phi(t) = \frac{\Pr(T'{=}t|\xi')}{\Pr(T{=}t|\xi)}.$$

Finally, we rewrite $\phi(t)$ to depend on the post-shift marginal distribution, $\Pr(T'{=}t)$, and the pre-shift conditional distributions, $\Pr(T{=}t|\xi)$ and $\Pr(C|T{=}t)$, for each $t \in \mathcal{T}$:

$$\begin{aligned}\phi(t) &= \frac{\Pr(T'{=}t|\xi')}{\Pr(T{=}t|\xi)} \\ &= \frac{\Pr(\xi'|T'{=}t)\Pr(T'{=}t)}{\Pr(T{=}t|\xi)\Pr(\xi')} && \textit{(Using Bayes' Theorem)} \\ &= \frac{\Pr(\xi|T{=}t)\Pr(T'{=}t)}{\Pr(T{=}t|\xi)\Pr(\xi')} && \textit{(Using (3))} \\ &= \frac{\Pr(\xi|T{=}t)\Pr(T'{=}t)}{\Pr(T{=}t|\xi)\sum_{t'\in\mathcal{T}}\Pr(\xi'|T'{=}t')\Pr(T'{=}t')} \\ &= \frac{\Pr(\xi|T{=}t)\Pr(T'{=}t)}{\Pr(T{=}t|\xi)\sum_{t'\in\mathcal{T}}\Pr(\xi|T{=}t')\Pr(T'{=}t').} && \textit{(Using (3))}\end{aligned}$$

□

D Enforcing Complex Fairness Definitions

In Section 3, we assume that the user-specified definition of unfair behavior, g, is defined by,

$$g(\theta) \coloneqq \mathbf{E}\big[\, H \,\big|\, \xi \,\big] - \tau. \tag{10}$$

However common fairness definitions such as the 80% rule (Griggs v. Duke Power Co., 1971) and equalized odds (Hardt et al., 2016) do not have the same form as (10). Nonetheless, these definitions can often be described by known *expressions* of multiple terms, where each term is a conditional or marginal expected value. Consider, for example, the definition of bias codified by *disparate impact* (Griggs v. Duke Power Co., 1971; Chouldechova, 2017; Zafar et al., 2017):

$$g_{\text{DI}}(\theta) \coloneqq 0.8 - \min\left\{\frac{\mathbf{E}[\theta(X)|S{=}\texttt{female}]}{\mathbf{E}[\theta(X)|S{=}\texttt{male}]}, \frac{\mathbf{E}[\theta(X)|S{=}\texttt{male}]}{\mathbf{E}[\theta(X)|S{=}\texttt{female}]}\right\}.$$

This definition is clearly not of the form described by (10); furthermore, it is challenging to estimate because i.i.d. samples of $\mathbf{E}[\theta(X)|S{=}\texttt{male}]$ and $\mathbf{E}[\theta(X)|S{=}\texttt{female}]$ cannot be simultaneously

computed given a single observation. Regardless, it is possible to compute valid high-confidence upper bounds on $g_{\text{DI}}(\theta)$ by leveraging the recursive bound-propagation described by Metevier et al. (2019). Importantly, we evaluate our proposed algorithms using complex but realistic fairness definitions by leveraging this strategy, and as a consequence, our results are influenced by this decision.

While we refer the reader to Metevier et al. (2019) for a more complete discussion of this strategy, we present a brief intuition here. First, we assume that the user's definition of g can be written as,

$$g(\theta) = f\big(\phi_1(\theta), ..., \phi_k(\theta)\big),$$

where each ϕ_i for $i \in \{1, ..., k\}$ denotes a *parameter* of the joint distribution of (X, Y, S), and f is some function of k arguments specified by the user. In our work, we assume that these parameters are each expressed as a conditional expected value analogous to (10) and that the expression defining f is provided by the user as text.

Following (Metevier et al., 2019), the expression for f is parsed into a tree structure representing the recursive application of various predefined operations. To construct a high-confidence upper bound on $g(\theta)$—or in this work, $g'(\theta)$—we first construct a set of confidence intervals on each parameter using the methods described in Section 3. Importantly, we apply the union bound to ensure that the set of confidence intervals on the parameters hold jointly with probability $1-\delta$. Next, these intervals are recursively propagated through the expression for f, where at each node of the computation tree, the interval for that node is computed by applying *interval arithmetic*, which describes the image of certain mathematical operations given intervals as their arguments. Since the root of the computation tree denotes the quantity $g'(\theta)$, the result of this recursive system is an interval containing the true value of $g'(\theta)$ with at least probability $1-\delta$.

A drawback of this approach is that it assumes that the intervals describing each parameter are independent, which is often false when considering demographic shift. While violation of this assumption does not impact the validity of the resulting confidence interval on $g'(\theta)$, it causes them to become larger than they would be if the dependence between each parameter were known. For example, suppose g is defined to measure bias based on sex, and consider two races. If, for individuals of one race, a certain sex is encountered much more often than other sexes, while for individuals of the second race, all sexes are encountered equally often, then a demographic shift that makes the first race more likely may cause certain parameters defining fairness to be highly correlated with others. By ignoring these dependencies, the approach presented in (Metevier et al., 2019) may produce significantly larger confidence intervals for $g'(\theta)$ compared to alternative approaches that leverage this dependency. For this reason, we consider this problem to be a strong candidate for future work, as it has the potential to improve the data efficiency of our methods as well as those proposed in existing work (Metevier et al., 2019; Thomas et al., 2019).

E Simulating and Evaluating Bounded Demographic Shift

Here, we describe our procedure for simulating the impact of demographic shift given a fixed population dataset when the exact deployment distribution is unknown. Intuitively, after generating a training dataset, we antagonistically select a new, non-uniform distribution over the population that satisfies the user's demographic shift assumptions—that is, that the marginal distribution over demographics is contained in $\mathcal{Q}$—but otherwise maximizes the prevalence of unfair behavior. Since the population and sampling distributions are known during evaluation, this oracle knowledge can be used to compute exact values for various statistics, such as expected classification accuracy and the value of $g'(\theta)$ for all models θ.

To make this procedure formal, let the population dataset be denoted by $\mathcal{D}_{pop}$:

$$\mathcal{D}_{pop} \coloneqq \{(x_i, y_i, s_i, t_i)\}_{i=1}^{n}.$$

Note that we do not refer to this set using the standard notation for random variables because in our experimental context the population is treated as a fixed, non-random population. To generate a random training dataset, D, we sample observations uniformly from $\mathcal{D}_{pop}$ with replacement. Specifically, if P denotes the uniform distribution over the observations in $\mathcal{D}_{pop}$, then training datasets are defined by $D \coloneqq \{(X_j, Y_j, S_j, T_j)\}_{j=1}^{n_0}$, where each $(X_j, Y_j, S_j, T_j) \sim P$.

Next, we generate a new distribution over the population that satisfies the user's assumptions but otherwise maximizes the prevalence of unfair behavior for a given model, which we denote by Q. However, to comply with the user's assumptions about the demographic shift, Q must be selected carefully. The following theorem provides the conditions that Q must satisfy to achieve this.

Theorem 2. *Let P denote a uniform distribution over $\mathcal{D}_{pop} \coloneqq \{(x_i, y_i, s_i, t_i)\}_{i=1}^n$. Assume that the demographic attribute takes values in some set $\mathcal{T}$ and that each demographic $t \in \mathcal{T}$ occurs at least once in the population. Next, let each $q \in \mathcal{Q}$ denote a marginal distribution over $\mathcal{T}$, where q_t denotes the probability of encountering demographic t. Finally, let $\mathbb{N}_{\mathcal{D}_{pop}}[x, y, s, t]$ denote the number of observations in $\mathcal{D}_{pop}$ that are equal to (x, y, s, t) and let $\mathbb{N}_{\mathcal{D}_{pop}}[t]$ denote the number of observations that have demographic attribute equal to t. It follows that Q, defined below, is a distribution over $\mathcal{D}_{pop}$ that satisfies both of the demographic shift assumptions, and has a marginal distribution over demographics that is contained in $\mathcal{Q}$:*

$$Q(X{=}x, Y{=}y, S{=}s, T{=}t) = \frac{\mathbb{N}_{\mathcal{D}_{pop}}[x, y, s, t]}{\mathbb{N}_{\mathcal{D}_{pop}}[t]} q_t.$$

Proof. To show this result, we derive an expression for Q that has these properties by construction.

First, we expand the post-shift joint distribution using the laws of conditional probability:

$$Q(X{=}x, Y{=}y, S{=}s, T{=}t) = Q(X{=}x, Y{=}y, S{=}s | T{=}t) Q(T{=}t).$$

Next, we apply the second demographic shift assumption:

$$Q(X{=}x, Y{=}y, S{=}s, T{=}t) = P(X{=}x, Y{=}y, S{=}s | T{=}t) Q(T{=}t).$$

Then, we represent the conditional $P(X, Y, S|T{=}t)$ as a ratio using laws of conditional probability:

$$Q(X{=}x, Y{=}y, S{=}s, T{=}t) = \frac{P(X{=}x, Y{=}y, S{=}s, T{=}t)}{P(T{=}t)} Q(T{=}t).$$

Because P is a uniform distribution over $\mathcal{D}_{pop}$, it follows that the value of $P(X{=}x, Y{=}y, S{=}s, T{=}t)$ is simply the number of occurrences of (x, y, s, t) in $\mathcal{P}$ divided by the total number of samples in the population, n. Similarly, $P(T{=}t)$ is equal to the number of observations that have demographic attribute equal to t, divided by n. Since we assume that each demographic is observed in the population, it follows that $P(T{=}t) > 0$ for all $t \in \mathcal{T}$. Let $\mathbb{N}_{\mathcal{D}_{pop}}[x, y, s, t]$ denote the number of observations in $\mathcal{D}_{pop}$ that are equal to (x, y, s, t) and let $\mathbb{N}_{\mathcal{D}_{pop}}[t]$ denote the number of observations that have demographic attribute equal to t. It follows that for all observations, $(x, y, s, t) \in \mathcal{D}_{pop}$, we have

$$Q(X{=}x, Y{=}y, S{=}s, T{=}t) = \frac{\mathbb{N}_{\mathcal{D}_{pop}}[x, y, s, t]}{\mathbb{N}_{\mathcal{D}_{pop}}[t]} Q(T{=}t).$$

Finally, we define the marginal distribution of Q over demographics to be given by q:

$$Q(X{=}x, Y{=}y, S{=}s, T{=}t) = \frac{\mathbb{N}_{\mathcal{D}_{pop}}[x, y, s, t]}{\mathbb{N}_{\mathcal{D}_{pop}}[t]} q_t.$$

Since Q has the same conditional distribution given the demographic as P by construction, it satisfies the demographic shift assumptions. Furthermore, since the marginal distribution of Q over demographics is defined to be given by q, which satisfies $q \in \mathcal{Q}$, it also satisfies the user's assumptions about the demographic shift. □

Theorem 2 shows that, given a $q \in \mathcal{Q}$, it is straightforward to construct a distribution over the population which satisfies the demographic shift assumptions. Therefore, to select a distribution that maximizes the prevalence of unfair behavior for a given model, θ, we numerically optimize $g'(\theta)$ over $q \in \mathcal{Q}$ using simplical homology optimization (Endres et al., 2018) to determine the maximizing marginal distribution, q^*, and then define the final distribution over $\mathcal{D}_{\text{pop}}$ using Theorem 2.

Theorem 2 can also be used to compute exact values for various statistics of interest during evaluation, such as expected classification accuracy or the value of $g'(\theta)$ for all models, θ. For example, consider

estimating the post-deployment classification accuracy of a model, θ, given by $\mathbf{E}_Q[\mathbb{I}[\theta(X)=Y]]$. If $\bar{\mathcal{D}}_{pop}$ denotes the set of unique observations in $\mathcal{D}_{pop}$, then we have

$$\begin{aligned}
&\mathbf{E}_Q\left[\,\mathbb{I}[\theta(X)=Y]\,\right] \\
&\quad= \sum_{(x,y,s,t)\in\bar{\mathcal{D}}_{pop}} \mathbb{I}[\theta(x)=y]\, Q(X=x, Y=y, S=s, T=t) \\
&\quad= \sum_{(x,y,s,t)\in\bar{\mathcal{D}}_{pop}} \mathbb{I}[\theta(x)=y] \frac{\mathbb{N}_{\mathcal{D}_{pop}}[x,y,s,t]}{\mathbb{N}_{\mathcal{D}_{pop}}[t]}\, Q(T=t).
\end{aligned}$$

Analogous expressions can be used to find exact values for post-shift model accuracy as well as the value of $g'(\theta)$ for all models θ.

F Additional Results for Bounded Demographic Shift

Results for our experiments on enforcing fairness under unknown demographic shift, according to the principles of equal opportunity, equalized odds, and predictive equality, are shown in Figure 6.

Demographic Parity (Dwork et al., 2012; Calders and Verwer, 2010)

$$g_{\text{DP}}(\theta) \coloneqq \left|\, \mathbf{E}[\theta(X)|S{=}\texttt{female}] - \mathbf{E}[\theta(X)|S{=}\texttt{male}] \,\right| - 0.1$$

Disparate Impact (Griggs v. Duke Power Co., 1971; Chouldechova, 2017; Zafar et al., 2017)

$$g_{\text{DI}}(\theta) \coloneqq 0.8 - \min\left\{ \frac{\mathbf{E}[\theta(X)|S{=}\texttt{female}]}{\mathbf{E}[\theta(X)|S{=}\texttt{male}]}, \frac{\mathbf{E}[\theta(X)|S{=}\texttt{male}]}{\mathbf{E}[\theta(X)|S{=}\texttt{female}]} \right\}$$

Equal Opportunity (Hardt et al., 2016; Chouldechova, 2017)

$$g_{\text{EOp}}(\theta) \coloneqq \left|\, \mathbf{E}[\theta(X)|Y{=}0, S{=}\texttt{female}] - \mathbf{E}[\theta(X)|Y{=}0, S{=}\texttt{male}] \,\right| - 0.1$$

Equalized Odds (Hardt et al., 2016)

$$g_{\text{EOd}}(\theta) \coloneqq \begin{array}{l} \left|\, \mathbf{E}[\theta(X)|Y{=}0, S{=}\texttt{female}] - \mathbf{E}[\theta(X)|Y{=}0, S{=}\texttt{male}] \,\right| + \\ \left|\, \mathbf{E}[1{-}\theta(X)|Y{=}1, S{=}\texttt{female}] - \mathbf{E}[1{-}\theta(X)|Y{=}1, S{=}\texttt{male}] \,\right| - 0.1 \end{array}$$

Predictive Equality (Chouldechova, 2017; Corbett-Davies et al., 2017)

$$g_{\text{PE}}(\theta) \coloneqq \left|\, \mathbf{E}[1{-}\theta(X)|Y{=}1, S{=}\texttt{female}] - \mathbf{E}[1{-}\theta(X)|Y{=}1, S{=}\texttt{male}] \,\right| - 0.1$$

Figure 5: Definitions of fairness used in the additional experiments presented in this section. These definitions were specified as text input and bounded with a recursive technique used by prior Seldonian algorithms (Metevier et al., 2019).

G Preliminary Experimental Results: UCI Adult Dataset, Known Demographic Shift

In this section, we include preliminary experimental results that will be included in Section 4. In these evaluations, we use the same experimental procedure as in Section 4.1 to simulate demographic shift and measure the accuracy of each model, as well as the frequency with which each training algorithm returns a model that violates fairness constraints.

These experiments were conducted using the UCI Adult data set, which includes various features, including race and sex, describing 48, 842 individuals taken during a 1994 US census (Kohavi and

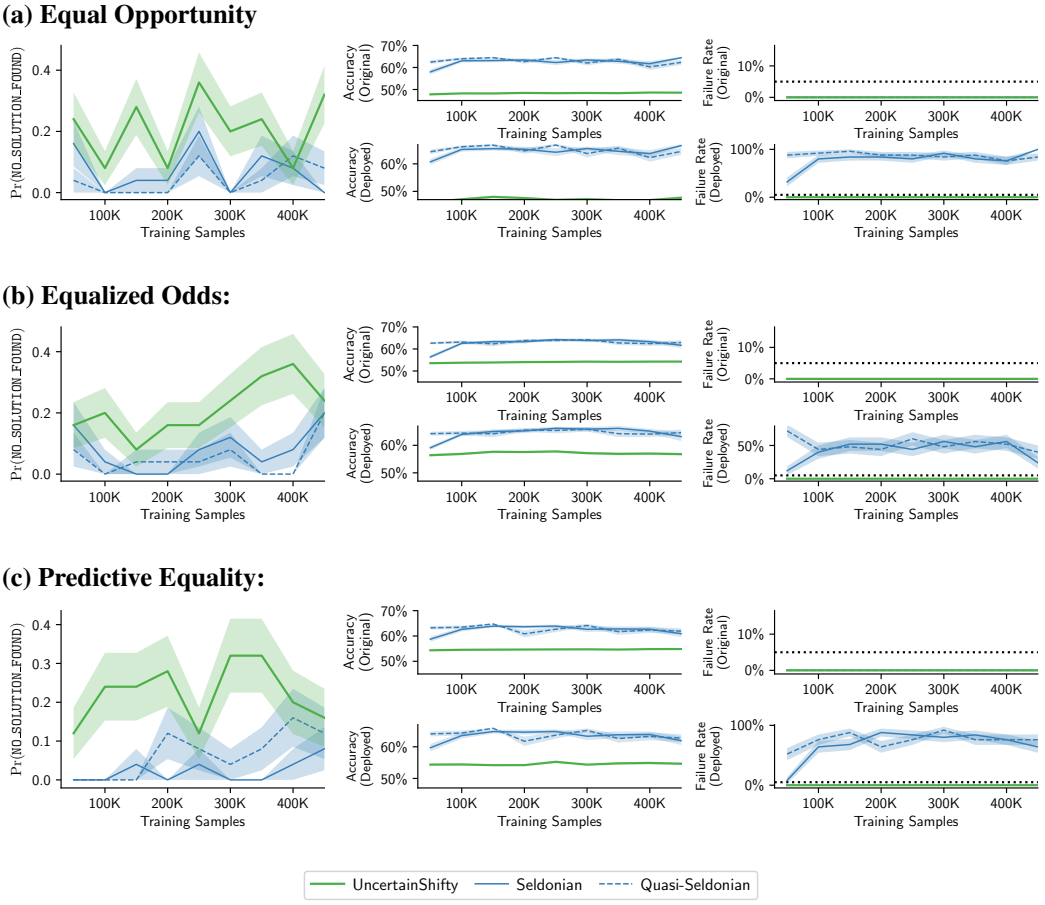


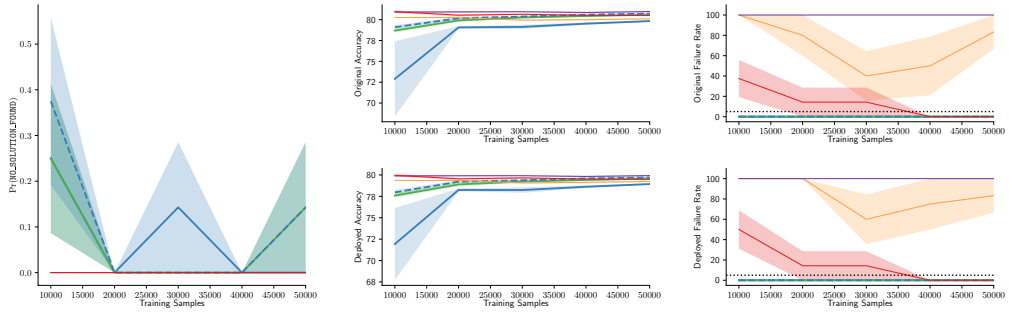
Figure 6: Results for experiments enforcing fairness constraints when the future marginal demographic distribution is unknown.

Becker, 1996). Given this dataset, we train classifiers to predict whether or not an individual earns above $50,000 each year. To assess fairness under demographic shift, we define the fairness attribute, S, to be the race of each individual, and define the demographic shift to be over T, the sex of each individual. Specifically, we consider the subset of the UCI Adult dataset corresponding to black or white individuals, and simulate a known demographic shift from the true population distribution in which $\Pr(T{=}\text{Female}) = 0.32$ during training while $\Pr(T'{=}\text{Female}) = 0.25$ during deployment.

We conducted trials using `Shifty-ttest`, two Seldonian algorithms, Fairness Constraints, and Fairlearn. In addition, we also evaluate *RFLearn*, which is an algorithm for training classifiers that promotes fair outcomes in the presence of *covariate shift*, but does not require data from the deployment distribution (Du and Wu, 2021). Importantly, RFLearn is not designed to address demographic shift, which has subtle differences from covariate shift as described in Appendix A, and promotes fairness based on demographic parity. While other methods have been proposed to promote fair outcomes under covariate shift, we were unable to compare to these approaches because they either assume access to data drawn from the deployment distribution (Rezaei et al., 2021; Coston et al., 2019), or they assume access to additional input such as causal graphs of the problem (Singh et al., 2021) in our experiments, which are unavailable in our experiments.

Due to time constraints in generating these preliminary results, we conducted 10 trials for each configuration, and trained each model using training sets with sizes ranging between 10,000 and 50,000 samples. Furthermore, we conducted our experiments using two definitions of fairness, demographic parity and predictive equality. Specifically, we used the definitions as shown in Figure 5,

(b) Demographic Parity:



(b) Predictive Equality:

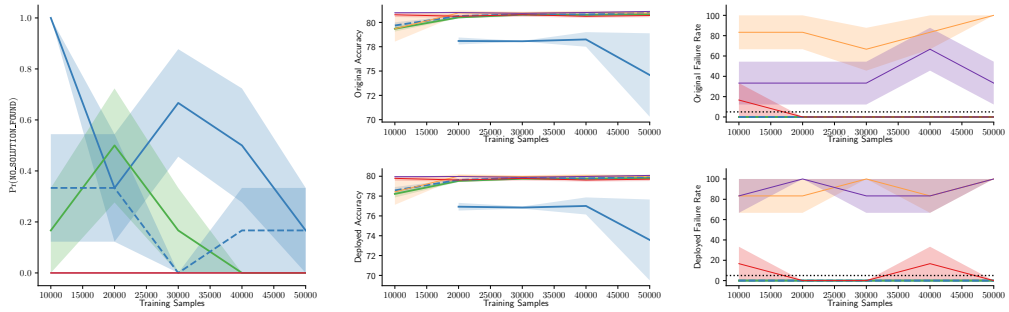


Figure 7: Preliminary results when enforcing fairness constraints under unknown demographic shift using the UCI Adult dataset.

with the exception that instead of defining S to be the sex of an individual, we defined S to be a binary random variable representing whether or not the individual's race was black or white.

Results for these experiments are shown in Figure 7. First, we found that in our experiments enforcing demographic parity and those enforcing predictive equality, `Shifty-ttest` satisfied Property B, as evidenced by the fact that the green curves in the rightmost plots of Figure 7 are never above the dashed black line that denotes the $\delta = 0.05$ tolerance. On the other hand, Fairlearn, Fairness Constraints, and RFLearn all produced unfair models frequently, although in our experiments using predictive equality, this happened only occasionally for Fairness Constraints. Interestingly, RFLearn, despite being designed to promote fairness with respect to demographic parity under covariate shift, consistently violated the demographic parity constraints set during our experiments both before and after deployment. Upon closer inspection of the results for demographic parity, we found that, on average, the value of $g'(\theta)$ for models trained using RFLearn was 0.0175, indicating that RFLearn identified models that were only slightly more unfair than the required tolerance. This highlights the observation that while many existing approaches to promoting fair ouctomes may find models that are reasonably fair, they do not necessarily provide guarantees that the prevalence of unfair behavior will meet specific tolerances with high confidence, as `Shifty` does.

Next, our results show that `Shifty-ttest` attained competitive accuracy compared to the best-performing baseline algorithms (middle column plots in Figure 7), despite the small amount of training data used in these evaluations compared to those in Section 4. Consequently, `Shifty-ttest` suffered very little loss in accuracy in order to enforce fairness constraints that hold under demographic shift for this experiment. Comparing these results to our results in Figure 3, this suggests that any accuracy loss exhibited by `Shifty-ttest` is dependent on the data distribution being learned, and may be negligible in some settings.

Finally, we note that, despite the small amount of training data used in these experiments, `Shifty-ttest` was able to frequently avoid returning `NO_SOLUTION_FOUND`.

FAIR REPRESENTATION LEARNING THROUGH IMPLICIT PATH ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

We considered a fair representation learning perspective, where optimal predictors, on top of the data representation, are ensured to be invariant with respect to different subgroups. Specifically, we formulated the problem as a bi-level optimization, where the representation is learned in the outer-level, and invariant optimal group predictors are updated in the inner-level. To avoid the high computational and memory cost of differentiating in the inner-level optimization, we proposed the implicit path alignment algorithm, which only relies on the solution of inner optimization and the implicit differentiation rather than the exact optimization path. Moreover, the proposed bi-level objective is demonstrated to fulfill the *sufficiency rule*, which is desirable in various practical scenarios but was not commonly studied in fair representation learning. We further analyzed the error gap of the implicit approach and empirically validated the proposed method in both classification and regression settings. Experimental results show the consistently better trade-off in prediction performance and fairness measurement.

1 INTRODUCTION

Machine learning has been widely adopted in the real world decision-making practice such as job candidate screening (Raghavan et al., 2020) and credit application. However, it has been observed that learning algorithms treated some groups of population unfavorably, for example, denying credit on the grounds of gender, age or ethnicity (Hardt et al., 2016). To this end, algorithmic fairness that is to mitigate the *prediction bias* for different subgroups has recently received tremendous attentions.

With the rapid advancement of representation learning (LeCun et al., 2015), learning a fair embedding (Zemel et al., 2013) has been recently highlighted. Specifically, the learned fair representation can easily transfer the unbiased prior knowledge to the downstream tasks, with various successful applications in computer vision (Kim et al., 2019; Kehrenberg et al., 2020), language understanding (Chang et al., 2019; Ethayarajh, 2020) and artificial intelligence for health (Fletcher et al., 2021). Typically, the fair representation learning is achieved by adding various statistical fair metrics during the training process.

Based on this, most existing fair representation approaches in classification or regression principally aim to meet the *independence* or *separation* rule, e.g., (Madras et al., 2018; Song et al., 2019; Chzhen et al., 2020). However, in various real-world scenarios, the *sufficiency rule* is preferable. For example, health systems rely on commercial algorithms to identify and help patients with complex health needs. The algorithm outputs a healthcare need score, where a higher score indicates the patient is sicker and requires more healthcare. Obermeyer et al. (2019) revealed that a widely used algorithm, typical of this industry-wide approach and affecting millions of patients, exhibits significant racial bias. At a given predicted healthcare need score $\hat{Y} = t$, Black patients are considerably sicker than White patients ($\mathbb{E}_{\text{black}}[Y|\hat{Y} = t] > \mathbb{E}_{\text{white}}[Y|\hat{Y} = t]$). Obermeyer et al. (2019) also pointed out that remedying the disparity would increase the percentage of Black patients receiv-

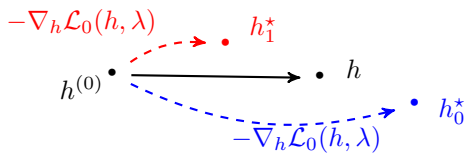
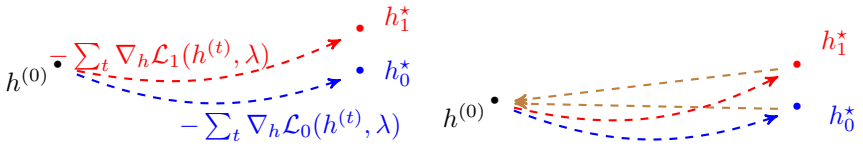


Figure 1: Unfair representation leads to different optimization path and non-invariant optimal predictors on the latent space $\mathcal{Z}$.



(a) Explicit Path (b) Implicit Path

Figure 2: Explicit and Implicit path alignment. (a) The considered fair representation learning criteria lies in ensuring the invariant optimal predictor w.r.t. different subgroups on $\mathcal{Z}$ ($h_0^\star = h_1^\star$). Since the gradient based approach is adopted to optimize h, the explicit path alignment aims to learn a representation λ to enforce the identical *optimization path* w.r.t. h. (b) The proposed implicit path alignment only requires the last iteration point and approximate the gradient w.r.t. λ from the last update of h (the brown arrow).

ing additional healthcare from 17.7 to 46.5%. Moreover, it has been theoretically justified (Barocas et al., 2019) that the Sufficiency rule is generally not compatible with Independence and Separation. Thus learning the fair representation w.r.t. the sufficiency rule is promising in both the algorithmic design and real-world applications.

In this paper, we address the sufficiency rule by considering the following intuition: given a fixed representation function, if the *optimal* predictor that learned on the embedding space are *invariant* from different sub-groups, then the corresponding representation function is fair. Fig. 1 provides an illustrative example. when the representation function $\lambda : \mathcal{X} \to \mathcal{Z}$ is unfair and we adopt gradient descent to learn the predictor $h : \mathcal{Z} \to R$. The optimal predictors of different subgroups (blue, red) are not invariant, resulting in biased predictions. We will later demonstrate such an intuition ensures the learned representation satisfying the sufficiency rule (Liu et al., 2019; Chouldechova, 2017).

The aforementioned intuition can be naturally formulated as a bi-level optimization problem, where we aim to adjust the representation λ (in the outer-level) to satisfy the invariant optimal predictor h (in the inner-level). Thus, when we adopt the gradient-based approach in solving the bi-level objective, a straightforward solution is to learn the representation λ to fulfill the identical *explicit* gradient-descent directions in learning predictor $h^\star$ of different groups, shown in Fig. 2(a). Intuitively, if the inner gradient descent step of each sub-group is identical, their final predictors (as the approximation of $h^\star$) will be invariant. However, the corresponding algorithmic realization is challenging in deep learning: 1) It requires storing the whole gradient steps, which induces a high memory burden. 2) the embedding function λ is optimized via backpropagation from the whole gradient optimization path, which induces a high computational complexity.

To this end, we propose an *implicit* path alignment, shown in Fig. 2(b). Notably, we only consider the final (t-th) update of the predictor $h^{(t)}$, then we update representation function λ by approximating its gradient at point $h^{(t)}$ through the implicit function (Bengio, 2000). By using the gradient approximation, it is no more required to store the whole gradient step and conduct the backpropagation through the entire path. Overall, the highlights in this paper are as follows:

Fair-representation learning to satisfy the sufficiency rule Instead of enforcing the independence or separation rule, the considered fair-representation criteria is proved to satisfy the sufficiency rule in both classification and regression. We also find such a criteria is intrinsically consistent with the recent Invariant Risk Minimization (IRM) (Arjovsky et al., 2019; Bühlmann, 2020), which aims to eliminate suspicious correlations while keeping robust correlations that are invariant across different environments. Intuitively, reducing the correlation w.r.t. the protected attributes enables the fair representation.

Principled and efficient algorithm We proposed a novel implicit path alignment algorithm to learn the fair representation, which addressed the prohibitive memory and computational cost in the original bi-level objective. Besides, we analyzed the approximation error gap of the proposed implicit algorithm, which induces a trade-off between the correct gradient estimation and fairness measures.

Improved fairness in classification and regression We evaluated the implicit algorithm in both classification and regression with tabular, computer vision and NLP datasets. Compared to the baselines, the implicit algorithm effectively improved the fairness with a smaller sufficiency gap.

2 PRELIMINARIES

We suppose the input $X \in \mathcal{X}$, the ground truth label $Y \in \mathcal{Y}$, and the algorithmic output $\hat{Y} \in \mathcal{Y}$. Throughout the paper, we only consider binary sensitive attribute (i.e, two sub-groups) with distributions $\mathcal{D}_0$ and $\mathcal{D}_1$. Then based on (Liu et al., 2019), the **sufficiency rule** is defined as:

$$\mathbb{E}_{\mathcal{D}_0}[Y|\hat{Y}=t] = \mathbb{E}_{\mathcal{D}_1}[Y|\hat{Y}=t], \ \ \forall t \in \mathcal{Y} \tag{1}$$

To measure the fairness w.r.t. the sufficiency rule, we propose the *sufficiency gap* as the metric. Since we aim to evaluate the fairness in both binary classification ($Y \in \{-1, 1\}$) and regression ($Y \in \mathbb{R}$), the metric is separately defined on these two scenarios.

Sufficiency gap in binary classification Based on the sufficiency rule, the sufficiency gap in binary classification is naturally defined as:

$$\Delta\text{Suf}_C = \sum_{y\in\{-1,1\}} |\mathcal{D}_0(Y=y|\hat{Y}=y) - \mathcal{D}_1(Y=y|\hat{Y}=y)| \tag{2}$$

ΔSuf_C encourages the two subgroups with identical Positive predicted value (PPV) and Negative predicted value (NPV). On the practical side, considering the healthcare evaluation system outputs either *High Risk* or *Low Risk*, Obermeyer et al. (2019) essentially revealed $\mathcal{D}_{\text{black}}(Y = \text{High Risk}|\hat{Y} = \text{Low Risk}) > \mathcal{D}_{\text{white}}(Y = \text{High Risk}|\hat{Y} = \text{Low Risk})$: the severity of Black patients is actually underestimated. Thus if ΔSuf_C is small, the racial discrimination can be remedied.

Sufficiency gap in regression Based on the sufficiency rule and (Kuleshov et al., 2018), the sufficiency gap in regression is defined as:

$$\Delta\text{Suf}_R = \int_{t\in\mathcal{Y}} |\mathcal{D}_0(Y \le t|\hat{Y} \le t) - \mathcal{D}_1(Y \le t|\hat{Y} \le t)|dt \tag{3}$$

$\Delta\text{Suf}_R \in [0, 1]$ is an approximation of $|\mathcal{D}_0(Y=y|\hat{Y}=y) - \mathcal{D}_1(Y=y|\hat{Y}=y)|$, $\forall y \in \mathbb{R}$, since the latter is difficult to estimate. From the practical aspect, assuming the health system outputs a *real-value* healthcare score $\hat{Y} = t$ (higher indicates sicker), Obermeyer et al. (2019); Sjoding et al. (2020) observed $\mathcal{D}_{\text{black}}(Y > t|\hat{Y} \le t) > \mathcal{D}_{\text{white}}(Y > t|\hat{Y} \le t)$: for the patients whose predicted healthcare score is less than t, the actual proportion of sicker ($Y > t$) in Black patients is considerably higher than White patients. Therefore a small ΔSuf_R suggests an improved disparity.

3 PROBLEM SETUP

We denote the representation function λ that maps the input X into the latent variable Z, the prediction function h such that $h : \mathcal{Z} \to \mathbb{R}$ for regression and $h : \mathcal{Z} \to \{-1, 1\}$ for binary classification. We then denote the prediction loss as ℓ, the prediction loss on subgroup $\mathcal{D}_0, \mathcal{D}_1$ is expressed as:

$$\mathcal{L}_0(h, \lambda) = \mathbb{E}_{(x,y)\sim\mathcal{D}_0}\ell(h \circ \lambda(x), y), \quad \mathcal{L}_1(h, \lambda) = \mathbb{E}_{(x,y)\sim\mathcal{D}_1}\ell(h \circ \lambda(x), y)$$

According to the intuition, we aim to solve the following bi-level objective:

$$\min_{\lambda} \ \mathcal{L}_0(h_0^\star, \lambda) + \mathcal{L}_1(h_1^\star, \lambda) \qquad \text{(Outer level)}$$

$$\text{s.t.} \ \ h_0^\star = h_1^\star, \ h_0^\star \in \operatorname*{argmin}_h \mathcal{L}_0(h, \lambda), \ h_1^\star \in \operatorname*{argmin}_h \mathcal{L}_1(h, \lambda). \qquad \text{(Inner level)}$$

Specifically, in the outer level, we aim to find a representation λ for minimizing the prediction error, given the optimal predictor $(h_0^\star, h_1^\star)$ on the embedding space $\mathcal{Z}$. As for the inner level, given a fixed representation λ, $h_0^\star$, $h_1^\star$ are the optimal predictor for each sub-group. The constraints $h_0^\star = h_1^\star$ additionally encourage the invariant optimal predictors from $\mathcal{D}_0$, $\mathcal{D}_1$.

Relation to the explicit path alignment In deep learning we adopt the gradient-based approach to minimize the loss, therefore $h^\star$ in the inner level is approximated as $h^{(t+1)}$, the t-th update in the gradient descent: $h_0^\star \approx h^{(0)} - \sum_t \nabla_h \mathcal{L}_0(h^{(t)}, \lambda)$, $h_1^\star \approx h^{(0)} - \sum_t \nabla_h \mathcal{L}_1(h^{(t)}, \lambda)$, where $h^{(0)}$ is the common initialization. Thus the invariant optimal predictor is equivalent to:

$$\sum_t \nabla_h \mathcal{L}_0(h^{(t)}, \lambda) = \sum_t \nabla_h \mathcal{L}_1(h^{(t)}, \lambda).$$

The aforementioned equation suggests learning a representation λ that ensures the identical optimization path w.r.t. h for each sub-group, which recovers the explicit path alignment.

Relation to the Sufficiency rule We further demonstrate the relation between the bi-level objective and Sufficiency rule.

Proposition 1. *If we specify the prediction loss ℓ as logistic regression loss in the classification $\log(1+\exp(-yh(z)))$ with $\mathcal{Y}=\{-1,1\}$ and the square loss in the regression $(h(z)-y)^2$ with $\mathcal{Y}\subset\mathbb{R}$. Then minimizing the inner-level loss is equivalent to:*

$$\mathbb{E}_{\mathcal{D}_0}[Y|Z=z]=\mathbb{E}_{\mathcal{D}_1}[Y|Z=z],\quad \mathbb{E}_{\mathcal{D}_0}[Y|\hat{Y}=h^\star(z)]=\mathbb{E}_{\mathcal{D}_1}[Y|\hat{Y}=h^\star(z)],$$

where $h^\star=h_0^\star=h_1^\star$ and $z=\lambda(x)$.

Proposition 1 reveals that the objective of inner-level loss is to encourage the sufficiency rule.

4 Practical Algorithms

In this section, we propose an implicit alignment in deep learning, where λ and h are implemented by the neural network. We also reformulate as the original objective through Lagrangian relaxation:

$$\min_{\lambda}\ \mathcal{L}_0(h_0^\star,\lambda)+\mathcal{L}_1(h_1^\star,\lambda)+\frac{\kappa}{2}\|h_0^\star-h_1^\star\|_2^2 \qquad \text{(Outer level)}$$

$$\text{s.t. } h_0^\star\in\operatorname*{argmin}_h \mathcal{L}_0(h,\lambda),\ h_1^\star\in\operatorname*{argmin}_h\mathcal{L}_1(h,\lambda), \qquad \text{(Inner level)}$$

where $\kappa>0$ is the coefficient to control the fairness. Then we will drive the approximated gradient w.r.t. λ, which contains the following key elements.

Solving the inner optimization Given a fixed representation λ, we find h_0^ϵ, h_1^ϵ such that:

$$\|h_0^\star-h_0^\epsilon\|\le\epsilon,\quad \|h_1^\star-h_1^\epsilon\|\le\epsilon,$$

where ϵ is the optimization tolerance. Besides, $h_1^\star$ and h_1^ϵ are essentially the function of λ, i.e., h_1^ϵ depends on the predefined representation function λ.

Computing the gradient of λ Given the approximate solution h_0^ϵ, h_1^ϵ, we can compute the gradient w.r.t. λ (referred as $\tilde{\text{grad}}(\lambda)$) [1] in the outer-level:

$$\begin{aligned}\tilde{\text{grad}}(\lambda)=&\nabla_\lambda\mathcal{L}_0(h_0^\epsilon,\lambda)+\left(\nabla_\lambda h_0^\epsilon\right)^T\left(\nabla_{h_0}\mathcal{L}_0(h_0^\epsilon,\lambda)+\kappa(h_0^\epsilon-h_1^\epsilon)\right)\\&+\nabla_\lambda\mathcal{L}_1(h_1^\epsilon,\lambda)+\left(\nabla_\lambda h_1^\epsilon\right)^T\left(\nabla_{h_1}\mathcal{L}_1(h_1^\epsilon,\lambda)-\kappa(h_0^\epsilon-h_1^\epsilon)\right).\end{aligned}$$

Where $\nabla_{h_0}\mathcal{L}_0(h_0^\epsilon,\lambda)$ is the partial derivative in the loss w.r.t. the first term (about h_0), evaluated at h_0^ϵ. Also $\nabla_\lambda\mathcal{L}_0(h_0^\epsilon,\lambda)$ is the partial derivative w.r.t. the second term (about λ).

Implicit function for approximating the gradient In order to compute $\tilde{\text{grad}}(\lambda)$ in `autograd`, we need to estimate $\nabla_\lambda h_0^\epsilon$ and $\nabla_\lambda h_1^\epsilon$. We herein adopt the implicit function (Bengio, 2000) to approximate $\nabla_\lambda h_0^\epsilon$, which has been adopted in the hyperparameter optimization (Pedregosa, 2016) and meta-learning (Rajeswaran et al., 2019).

Concretely, if the prediction loss is smooth and there exist stationary points to achieve optimal, we have: $\nabla_{h_0}\mathcal{L}_0(h_0^\star(\lambda),\lambda)=0,\nabla_{h_1}\mathcal{L}_0(h_1^\star(\lambda),\lambda)=0$. Then differentiating w.r.t. λ will induce: $\mathbf{d}\left(\nabla_{h_0}\mathcal{L}_0(h_0^\star(\lambda),\lambda)\right)/\mathbf{d}\lambda=\nabla^2_{h_0}\mathcal{L}_0(h_0^\star,\lambda)\nabla_\lambda h_0^\star+\nabla_\lambda\nabla_{h_0}\mathcal{L}_0(h_0^\star,\lambda)=0$.[2] Thus we have $\nabla_\lambda h_0^\star=-\left(\nabla^2_{h_0}\mathcal{L}_0(h_0^*,\lambda)\right)^{-1}\left(\nabla_\lambda\nabla_{h_0}\mathcal{L}_0(h_0^*,\lambda)\right)$, where the Hessian matrix $\nabla^2_{h_0}\mathcal{L}_0(h_0^*,\lambda)$ is assumed to be invertible.

Through the implicit function, we can approximate $\nabla_\lambda h_0^\epsilon$ as:

$$\nabla_\lambda h_0^\epsilon\approx-\left(\nabla^2_{h_0}\mathcal{L}_0(h_0^\epsilon,\lambda)\right)^{-1}\left(\nabla_\lambda\nabla_{h_0}\mathcal{L}_0(h_0^\epsilon,\lambda)\right)$$

As for $\nabla_\lambda h_1^\epsilon$, we have the similar result: $\nabla_\lambda h_1^\epsilon\approx-\left(\nabla^2_{h_1}\mathcal{L}_1(h_1^\epsilon,\lambda)\right)^{-1}\left(\nabla_\lambda\nabla_{h_1}\mathcal{L}_1(h_1^\epsilon,\lambda)\right)$.

[1] We denote the ground truth gradient as $\text{grad}(\lambda)$ if we adopt optimal predictor $h_0^\star,h_1^\star$ in the computation.

[2] $\mathbf{d}(\cdot)/\mathbf{d}\lambda$ denotes the total derivative.

Algorithm 1 Implicit Path Alignment Algorithm

Ensure: Representation function λ, predictor h_0, h_1, datasets from two sub-groups $\mathcal{D}_0$. $\mathcal{D}_1$.
1: **for** mini-batch of samples from $(\mathcal{D}_0, \mathcal{D}_1)$ **do**
2: Solving the inner-level optimization with tolerance ϵ. Obtaining $h_0^\epsilon, h_1^\epsilon$.
3: Solving Eq. (4) with tolerance δ. Obtaining $\mathbf{p}_0^\delta$ and $\mathbf{p}_1^\delta$.
4: Computing $\tilde{\text{grad}}^\delta(\lambda)$ (gradient of representation λ)
5: Updating λ through `autograd`: $\lambda \leftarrow \lambda - \tilde{\text{grad}}^\delta(\lambda)$
6: **end for**
7: **return** λ, h_0^ϵ, h_1^ϵ

Efficient and numerical stable gradient estimation Plugging in the approximations, the gradient w.r.t λ is approximated as:

$$\tilde{\text{grad}}(\lambda) \approx \nabla_\lambda \mathcal{L}_0(h_0^\epsilon, \lambda) - (\nabla_\lambda \nabla_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda))^T \underbrace{\left(\nabla^2_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda)\right)^{-1} (\nabla_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda) + \kappa(h_0^\epsilon - h_1^\epsilon))}_{\mathbf{p}_0}$$
$$+ \nabla_\lambda \mathcal{L}_1(h_1^\epsilon, \lambda) - (\nabla_\lambda \nabla_{h_1} \mathcal{L}_1(h_1^\epsilon, \lambda))^T \underbrace{\left(\nabla^2_{h_1} \mathcal{L}_1(h_1^\epsilon, \lambda)\right)^{-1} (\nabla_{h_1} \mathcal{L}_1(h_1^\epsilon, \lambda) - \kappa(h_0^\epsilon - h_1^\epsilon))}_{\mathbf{p}_1}$$

However, the current form is still computationally expensive due to the computation of inverse Hessian matrix. To this end, we denote $\mathbf{p}_0$ and $\mathbf{p}_1$ as the inverse-Hessian vector product. Then computing $\mathbf{p}_0$ and $\mathbf{p}_1$ is equivalent to solve the following quadratic programming (QP):

$$\begin{aligned} &\text{argmin}_{\hat{\mathbf{p}}_0} \frac{1}{2} \hat{\mathbf{p}}_0^T \left(\nabla^2_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda)\right) \hat{\mathbf{p}}_0 - \hat{\mathbf{p}}_0^T \left(\nabla_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda) + \kappa(h_0^\epsilon - h_1^\epsilon)\right) \\ &\text{argmin}_{\hat{\mathbf{p}}_1} \frac{1}{2} \hat{\mathbf{p}}_1^T \left(\nabla^2_{h_1} \mathcal{L}_1(h_1^\epsilon, \lambda)\right) \hat{\mathbf{p}}_1 - \hat{\mathbf{p}}_1^T \left(\nabla_{h_1} \mathcal{L}_1(h_1^\epsilon, \lambda) - \kappa(h_0^\epsilon - h_1^\epsilon)\right) \end{aligned} \tag{4}$$

Since it is a typical QP problem and we adopt conjugate gradient method (Concus et al., 1985; Rajeswaran et al., 2019), which can be updated efficiently through `autograd` via computing the Hessian-vector product. We additionally suppose the optimization error in the QP as δ, i.e.: $\|\mathbf{p}_0 - \mathbf{p}_0^\delta\| \leq \delta$, $\|\mathbf{p}_1 - \mathbf{p}_1^\delta\| \leq \delta$, then the gradient w.r.t representation λ can be finally expressed as:

$$\tilde{\text{grad}}^\delta(\lambda) = \nabla_\lambda \mathcal{L}_0(h_0^\epsilon, \lambda) - (\nabla_\lambda \nabla_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda))^T \mathbf{p}_0^\delta + \nabla_\lambda \mathcal{L}_1(h_1^\epsilon, \lambda) - (\nabla_\lambda \nabla_{h_1} \mathcal{L}_1(h_1^\epsilon, \lambda))^T \mathbf{p}_1^\delta$$

The $\tilde{\text{grad}}^\delta(\lambda)$ can be also efficiently estimated through Hessian vector product via `autograd` without explicitly computing the Hessian matrix.

Proposed algorithm Based on the key elements, the proposed algorithm is shown in Algo. 1.

4.1 The cost of Implicit algorithm: Approximation-Fair Trade-Off

Theorem 1 (Approximation Error Gap)**.** *Suppose that (1)* Smooth Predictive Loss*. The first-order derivatives and second-order derivatives of $\mathcal{L}$ are Lipschitz continuous; (2)* Non-singular Hessian matrix*. We assume $\nabla_{h_0,h_0} \mathcal{L}_0(h_0, \lambda), \nabla_{h_1,h_1} \mathcal{L}_1(h_1, \lambda)$, the Hessian matrix of the inner optimization problem, are invertible. (3)* Bounded representation and predictor function*. We assume the λ and h are bounded, i.e., $\|\lambda\|, \|h\|$ are upper bounded by the predefined positive constants. Then the approximation error between the ground truth and algorithmic estimated gradient w.r.t. the representation is be upper bounded by:*

$$\|grad(\lambda) - \tilde{grad}^\delta(\lambda)\| = \mathcal{O}(\kappa\epsilon + \epsilon + \delta).$$

The proof is delegated in Appendix B. We also discuss the assumptions to guarantee the convergence of Algorithm 1, shown in Appendix C.

Theorem 1 reveals that the gradient approximation error depends on the two-level optimization tolerance ϵ, δ and the coefficient of fair constraints κ. Specifically, the error gap reveals the inherent trade-off in accurate gradient estimation and fair-representation learning. If we fix the optimization tolerance ϵ and δ, a smaller κ indicates a better approximation of the gradient, which yields weak fair constraints. Thus the implicit alignment introduces a trade-off in the prediction performance (i.e., correct approximation of the gradient) and fairness measurement.

5 Related Work

Fair Machine Learning Below we only list the most related work in the *fairness* and refer to the survey paper (Mehrabi et al., 2021) for details in the algorithmic fairness. In the *classification*, various methods in learning fair representations have been proposed. Specifically, a common strategy is to introduce the statistical constraints as the regularization during the training, e.g., demographic parity (DP) (Zhang et al., 2018; Madras et al., 2018; Song et al., 2019; Jiang et al., 2020; Kehrenberg et al., 2020) or equalized odds (EO) (Song et al., 2019; Gupta et al., 2021) as the proxy of the separation and independence rule. Another direction is to disentangle the data for factorizing meaningful representations such as (Locatello et al., 2019). Intuitively, the disentangled embedding is independent of the sensitive attribution, thus reflecting a fair representation w.r.t. the independence rule, which can be potentially problematic when the label distributions of subgroups vary dramatically (Zhao et al., 2019).

Fairness has also been extended to the fields beyond classification. For instance, in the *regression* problem (Komiyama et al., 2018; Agarwal et al., 2019), the bounded group loss has been proposed as the fair measure: if prediction loss in each subgroup is smaller than ϵ, the regression is ϵ-level fair. In fact, the fair criteria in our paper is *not equivalent* to ϵ-fair. Given a fixed λ, the ϵ-level fair does not guarantee the *optimal* and *invariant* predictor for each subgroup and vice versa.

The sufficiency rule has also been discussed in the previous work. Notably, Chouldechova (2017); Liu et al. (2019) proposed the sufficiency gap *in classification* for measuring fairness w.r.t. the sufficiency rule. Liu et al. (2019) also discussed the inequivalence between the sufficiency gap and probabilistic calibration (Guo et al., 2017) (referred as calibration gap). According to Pleiss et al. (2017), the calibration rule is a stronger condition than sufficiency rule while it simultaneously hurts the prediction performance. Throughout this paper, we only consider the sufficiency rule. The triple trade-off between the calibration rule, sufficiency rule, and prediction performance will be left as future work.

Invariant Risk Minimization The analyzed fair-representation criteria shares a quite similar spirit to the IRM (Arjovsky et al., 2019; Bühlmann, 2020; Creager et al., 2021), where an algorithm IRM_v1 is proposed to enable the out-of-distribution (OOD) generalization. The key difference between our work and (Arjovsky et al., 2019) lies in the algorithmic aspect: it has been theoretically justified that the originally proposed IRM_v1 does not necessarily capture the invariance (Rosenfeld et al., 2020). By contrast, we directly solve the bi-level objective in the context of deep-learning and propose an efficient practical algorithm with better empirical performance than IRM_v1. Besides, based on Chen et al. (2021), the proposed algorithm does not provably guarantee the OOD generalization property due to the limited subgroups ($N = 2$) considered within the paper.

6 Experiments

6.1 Experimental setup

In the paper, we adopt the sufficiency gap as fair metrics, where $\hat{Y}$ is denoted as:

$$\hat{Y} = \begin{cases} h_0^\epsilon \circ \lambda(X), & X \in \mathcal{D}_0 \\ h_1^\epsilon \circ \lambda(X), & X \in \mathcal{D}_1 \end{cases}$$

Then in the binary classification, we can estimate $\Delta\text{Suf}_C = \sum_{y\in\{-1,+1\}} |\mathcal{D}_0(Y = y|\hat{Y} = y) - \mathcal{D}_1(Y = y|\hat{Y} = y)|$ from the data.

As for regression, the sufficiency gap $\Delta\text{Suf}_R = \int_t |\mathcal{D}_0(Y \le t|\hat{Y} \le t) - \mathcal{D}_1(Y \le t|\hat{Y} \le t)|$ (shown in Fig. 3, the orange region) is difficult to estimate due to the integration. To address this, we sample multiple values $\{t_1, \ldots, t_m\}$ and compute its average difference as the approximation of the integration. $\Delta\text{Suf}_R \approx \frac{1}{m}\sum_{i=1}^m |\mathcal{D}_0(Y \le t_i|\hat{Y} \le t_i) - \mathcal{D}_1(Y \le t_i|\hat{Y} \le t_i)|$

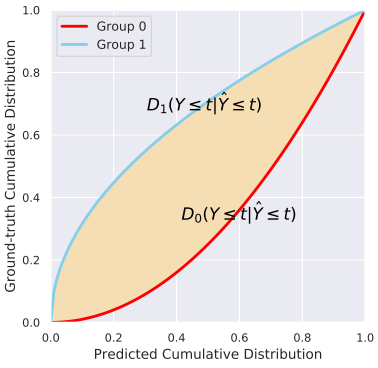


Figure 3: Sufficiency gap (ΔSuf_R) in regression

Method	Accuracy (↑)	ΔSuf_C (↓)
ERM (I)	0.768 ± 0.004	0.173 ± 0.008
Adv_debias (II)	0.760 ± 0.008	0.291 ± 0.006
Mixup (III)	0.758 ± 0.003	0.343 ± 0.022
IRM_v1 (IV)	0.753 ± 0.004	0.057 ± 0.015
One_step (V)	0.755 ± 0.007	0.048 ± 0.008
Implicit	0.760 ± 0.007	0.051 ± 0.012

Table 1: Toxic comments dataset. Accuracy and ΔSuf_C in different approaches.

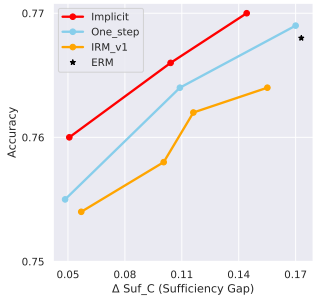


Figure 4: Toxic. Accuracy-Fair Trade-off

Concretely, for a given t_i in each group, we compute the percentile ($\hat{Y}_0$) at point t: $\mathcal{D}_0(\hat{Y}_0 \leq t_i)$, then we compute the corresponding ground truth cumulative distribution (Y) at the same point t_i: $\mathcal{D}(Y \leq t_i|\hat{Y} \leq t_i)$. Through the aforementioned approximation, we can compute $|\mathcal{D}_0(Y \leq t_i|\hat{Y} \leq t_i) - \mathcal{D}_1(Y \leq t_i|\hat{Y} \leq t_i)|$.

Baselines We consider the baselines that add fairness constraints during the training process. Specifically, we compare our method with (I) empirical risk minimization (ERM) that trains the model without considering fairness; (II) adversarial debiasing (Zhang et al., 2018); (III) fair mix-up (Chuang & Mroueh, 2021), a recent data-augmentation and effective approach in the fair representation learning. In fact, the baselines (II) and (III) are DP-based fair approaches, which is designed to demonstrate the general **non-compatibility** in addressing the sufficiency based fairness.

Besides, we include two additional baselines that have the similar objective but different algorithmic realizations. (IV) the original IRM regularization (referred as IRM_v1) (Arjovsky et al., 2019), which adds a gradient penalty to encourage the invariance. (V) One-step explicit alignment. In the inner-level optimization, we suppose to conduct the one-step gradient descent for each sub-group. Then in the outer-level optimization, we add a gradient-incoherence constraint to encourage the identical (one-step) optimization path: $\min_\lambda \|\nabla_{h_0}\mathcal{L}_0(h_0, \lambda) - \nabla_{h_1}\mathcal{L}_1(h_1, \lambda)\|_2^2$. All the results are reported by averaging five repetitions and additional experimental details are delegated in the Appendix.

6.2 Empirical Results

6.2.1 Toxic Comments

The toxic comments dataset (Jigsaw, 2018) is a binary **classification** task in NLP to predict whether comment is toxic or not. The original label is actually not binary since the comments is decided by multiple annotators, where the labelling discrepancy generally occurs. To this end, we conduct a simple strategy to decide comment is toxic if at least one annotator marks it. In this dataset, a portion of comments have been labeled with identity attributes, including gender and race. It has also been revealed that the race identity (e.g., black) is correlated with the toxicity label, which can lead to the predictive discrimination. Thus we adopted the *race* as the protected group by selecting two subgroups of Black and Asian. For the sake of computational simplicity, we first applied the pretrained BERT (Devlin et al., 2018) to extract the word embedding with 748 dimensional vector. Then we adopt representation function λ as two fully-connected layers with hidden dimension 200 with Relu activation and classifier h as a linear predictor. We report the test-set sub-group average accuracy and sufficiency gap (ΔSuf_C) in Tab. 1 and Fig. 4.

The results reveal several interesting facts. (1) The Demographic Parity (DP) based fair constraints are generally non-compatible with the sufficiency rule. Specifically, baseline (II,III) even increase ΔSuf_C with higher value than ERM. (2) For the baselines that track the sufficiency rule (IV,V), the sufficiency gap ΔSuf_C is improved with a similar accuracy, shown in Tab.1. We also change the regularization coefficient in (IV,V) and κ in the implicit approach. We observe that the implicit approach demonstrates a consistent better Accuracy-Fair trade-off, shown in Fig. 4.

Method	Accuracy (↑)	ΔSuf_C (↓)
ERM (I)	0.780 ± 0.015	0.210 ± 0.022
Adv_debias (II)	0.785 ± 0.022	0.165 ± 0.028
Mixup (III)	0.792 ± 0.011	0.160 ± 0.010
IRM_v1 (IV)	0.795 ± 0.012	0.086 ± 0.015
One_step (V)	0.797 ± 0.006	0.086 ± 0.012
Implicit	0.794 ± 0.027	0.074 ± 0.020

Table 2: CelebA dataset. Accuracy and predictive parity in different approaches.

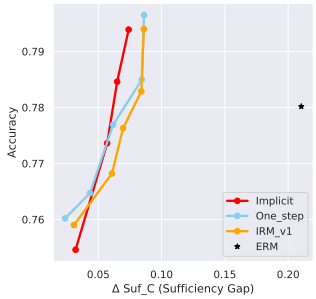


Figure 5: CelebA. Accuracy-Fair Trade-off

6.2.2 CELEBA DATASET

The CelebA dataset (Liu et al., 2015) contains around 200K images of celebrity faces, where each image is associated with 40 human-annotated binary attributes including gender, hair color, young, etc. In this paper, we designate *gender* as the sensitive attribute, and *attractive* as the binary **classification** task. We randomly select around 82K and 18K images as the training and validation set. Then we adopt representation function λ as pre-trained ResNet-18 (He et al., 2016) and classifier h as two-fully connected layers. We report the test-set sub-group average accuracy and sufficiency gap (ΔSuf_C) in Tab. 2 and Fig. 5.

The results in the CelebA show similar behaviors with the Toxic comments. Specifically, the DP based fair approaches (II, III) did not effectively improve ΔSuf_C, shown in Tab. 2. In contrast, the sufficiency can be significantly improved in baselines (IV, V) and implicit approach without largely losing the accuracy. Specifically, Fig. 5 visualizes the accuracy-fair trade-off curve, where the later three approaches show quite similar behaviors.

6.2.3 LAW DATASET

The Law Dataset is a **regression** task to predict a students GPA (real value, ranging from $[0, 4]$), where the data is utilized from the School Admissions Councils National Longitudinal Bar Passage Study (Wightman, 1998) with 20K examples. In the regression task, we adopt the square loss and *race* as the protected attribute (white versus non-white). We adopt λ as the one fully connected layer with hidden dimension 100 and Relu activation and predictor h as a linear predictor. We report the test-set subgroup average MSE (Mean Square Error) and sufficiency gap (ΔSuf_R) in Tab. 1 and Fig. 4.

Compared to the classification task, the results show similar behaviors in the regression. Specifically, the DP based fair approaches (II, III) still increase ΔSuf_R in the regression. In contrast, the gap is significantly improved in our proposed approach and baseline (IV,V). Specifically, Fig. 7 visualizes the sufficiency-gap of different approaches, where the implicit approach significantly mitigate the sufficiency gap. Besides, Fig. 6 shows the MSE-sufficiency gap curve, which still reveals the implicit approach benefits a better trade-off between the performance and fairness.

Method	MSE (↓)	ΔSuf_R (↓)
ERM (I)	0.190 ± 0.005	0.160 ± 0.007
Adv_debias (II)	0.223 ± 0.008	0.188 ± 0.012
Mixup (III)	0.216 ± 0.012	0.172 ± 0.007
IRM_v1 (IV)	0.208 ± 0.006	0.096 ± 0.006
One_step (V)	0.204 ± 0.007	0.125 ± 0.010
Implicit	0.198 ± 0.005	0.091 ± 0.011

Table 3: Law dataset. MSE and sufficiency gap in different approaches.

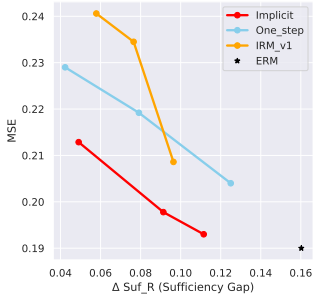
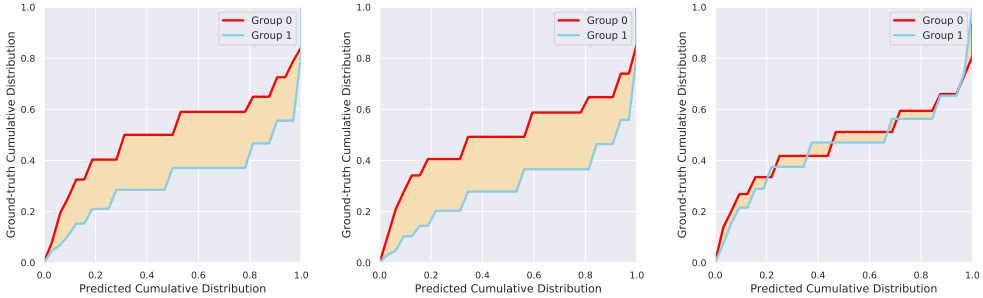


Figure 6: Law. MSE-Fair Trade-off



(a) ERM (b) Fair Mix-up (c) Implicit

Figure 7: Illustration of the sufficiency gap (ΔSuf_R) in Law dataset (regression). The ERM and Fair mix-up suffer a high ΔSuf_R, while the proposed implicit alignment can significantly mitigate the sufficiency gap.

Method	MSE (↓)	ΔSuf_R (↓)
ERM (I)	1.939 ± 0.021	0.246 ± 0.019
Adv_debias (II)	1.982 ± 0.016	0.252 ± 0.020
Mixup (III)	1.979 ± 0.025	0.246 ± 0.023
IRM_v1 (IV)	1.927 ± 0.031	0.077 ± 0.009
One_step (V)	1.904 ± 0.027	0.090 ± 0.019
Implicit	1.906 ± 0.019	0.051 ± 0.005

Table 4: NLSY dataset. MSE and sufficiency gap in different approaches.

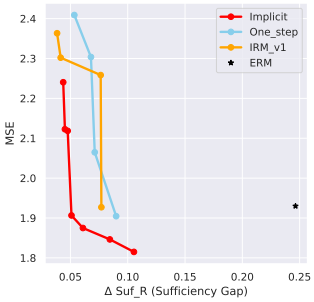


Figure 8: NLSY. MSE-Fair Trade-off

6.2.4 NLSY Dataset

The National Longitudinal Survey of Youth (NLSY, 2021) dataset is a **regression** task with around 7K dataset, which involves the survey results of the U.S. Bureau of Labor Statistics. It is intended to gather information on the labor market activities and other life events of several groups for predicting the income y of each person. We treat the *gender* as the sensitive attribute. We also normalize the output y by diving the $10,000$, then the final output y ranges around $[0, 8]$. The prediction loss is also the square loss. We adopt representation λ as the two fully connected layers with hidden dimension 200 and Relu activation and predictor h as a linear predictor. We report the test-set sub-group average MSE (Mean Square Error) and Sufficiency Gap (ΔSuf_R) in Tab. 4 and Fig. 8.

Tab. 4 provides similar trends with other datasets. Baselines (IV,V) and implicit approach effective control the sufficiency gap, while the DP based approach generally fails to improve the gap. Fig. 8 reveals a slightly better approximation-fair trade off for the implicit approach. Finally, Fig. 11 (in Appendix) visualizes the sufficiency gap of different algorithms. The gap is actually significantly improved while the calibration gap still exists, which is consistent with (Liu et al., 2019). Therefore it can be quite interesting and promising to analyze the triple trade-off between the sufficiency gap, calibration gap and prediction performance in the regression.

7 Conclusion

We considered the fair representation learning from a novel perspective through encouraging the invariant optimal predictors on the top of data representation. Then we formulated this problem as a bi-level optimization and proposed an implicit alignment algorithm. We further demonstrated the bi-level objective is to fulfil the sufficiency rule. Besides, we also analyzed the error gap of the implicit algorithm. The empirical results in both classification and regression settings suggest the improved fairness measurement. Finally, we think the future work can include developing computationally efficient explicit algorithms for avoiding the biased gradient computation.

ETHICS STATEMENT

This paper proposed a novel fair representation algorithm, which aims to address the potential prediction discrimination towards several subgroups. The proposed approach may also introduce the potential negative impact: we merely address the fairness with respect to the sufficiency rule in the paper, which is not always the preferable criteria in several specific scenarios.

REPRODUCIBILITY STATEMENT

We provided a demo source code in the supplementary material for a better understanding the proposed algorithm. Besides, the detailed experimental descriptions and theoretical proofs are also provided in the appendix.

REFERENCES

Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pp. 120–129. PMLR, 2019.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Peter Bühlmann. Invariance, causality and robustness. *Statistical Science*, 35(3):404–426, 2020.

Kai-Wei Chang, Vinod Prabhakaran, and Vicente Ordonez. Bias and fairness in natural language processing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*, Hong Kong, China, November 2019. Association for Computational Linguistics. URL `https://aclanthology.org/D19-2004`.

Yining Chen, Elan Rosenfeld, Mark Sellke, Tengyu Ma, and Andrej Risteski. Iterative feature matching: Toward provable domain generalization with logarithmic environments. *arXiv preprint arXiv:2106.09913*, 2021.

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

Ching-Yao Chuang and Youssef Mroueh. Fair mixup: Fairness via interpolation. *arXiv preprint arXiv:2103.06503*, 2021.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Fair regression with wasserstein barycenters. *arXiv preprint arXiv:2006.07286*, 2020.

P. Concus, G. Golub, and Gérard Meurant. Block preconditioning for the conjugate gradient method. *Siam Journal on Scientific and Statistical Computing*, 6, 01 1985. doi: 10.1137/0906018.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pp. 2189–2200. PMLR, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Kawin Ethayarajh. Is your classifier actually biased? measuring fairness under uncertainty with bernstein bounds. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2914–2919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.262. URL https://aclanthology.org/2020.acl-main.262.

Richard Ribón Fletcher, Audace Nakeshimana, and Olusubomi Olubeko. Addressing fairness, bias, and appropriate use of artificial intelligence and machine learning in global health. *Frontiers in Artificial Intelligence*, 3:116, 2021. ISSN 2624-8212. doi: 10.3389/frai.2020.561802. URL https://www.frontiersin.org/article/10.3389/frai.2020.561802.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Umang Gupta, Aaron Ferber, Bistra Dilkina, and Greg Ver Steeg. Controllable guarantees for fair outcomes via contrastive information estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7610–7619, 2021.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ray Jiang, Aldo Pacchiano, Tom Stepleton, Heinrich Jiang, and Silvia Chiappa. Wasserstein fair classification. In *Uncertainty in Artificial Intelligence*, pp. 862–872. PMLR, 2020.

Jigsaw. Toxic comment classification challenge, 2018. URL https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview/description.

Thomas Kehrenberg, Myles Bartlett, Oliver Thomas, and Novi Quadrianto. Null-sampling for interpretable and fair representations. In *European Conference on Computer Vision*, pp. 565–580. Springer, 2020.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9012–9020, 2019.

Junpei Komiyama, Akiko Takeda, Junya Honda, and Hajime Shimao. Nonconvex optimization for regression with fairness constraints. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2737–2746. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/komiyama18a.html.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pp. 2796–2804. PMLR, 2018.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Lydia T. Liu, Max Simchowitz, and Moritz Hardt. The implicit fairness criterion of unconstrained learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4051–4060. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/liu19f.html.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Francesco Locatello, Gabriele Abbati, Tom Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. On the fairness of disentangled representations. *arXiv preprint arXiv:1905.13662*, 2019.

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

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

NLSY. National longitudinal survey of youth, 2021. URL `https://www.bls.gov/nls/`.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019. doi: 10.1126/science.aax2342. URL `https://www.science.org/doi/abs/10.1126/science.aax2342`.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pp. 737–746. PMLR, 2016.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *arXiv preprint arXiv:1709.02012*, 2017.

Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 469–481, 2020.

Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in neural information processing systems*, 2019.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

Michael W Sjoding, Robert P Dickson, Theodore J Iwashyna, Steven E Gay, and Thomas S Valley. Racial bias in pulse oximetry measurement. *New England Journal of Medicine*, 383(25):2477–2478, 2020.

Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2164–2173. PMLR, 2019.

Linda F. Wightman. Lsac national longitudinal bar passage study, 1998.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pp. 325–333. PMLR, 2013.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018.

Wenbin Zhang and Eirini Ntoutsi. Faht: an adaptive fairness-aware decision tree classifier. *arXiv preprint arXiv:1907.07237*, 2019.

Han Zhao, Amanda Coston, Tameem Adel, and Geoffrey J Gordon. Conditional learning of fair representations. *arXiv preprint arXiv:1910.07162*, 2019.

A PROPOSITION 1

We consider the regression and classification separately.

Regression According to the definition, given a fixed and deterministic representation λ, we have

$$\mathcal{L}_0(h, \lambda) = \mathbb{E}_{\mathcal{D}_0}(h(z) - y)^2$$

It is noted as a typical regression problem with square error. We set the derivative as zero: $\nabla_h \mathcal{L}_0(h, \lambda) = 0$, we have $h_0^\star(z) = \mathbb{E}_{\mathcal{D}_0}[Y|Z = z]$. As for $\mathcal{D}_1$, we apply the same strategy with $h_1^\star(z) = \mathbb{E}_{\mathcal{D}_1}[Y|Z = z]$. Based on the invariant optimal predictor, we have $\mathbb{E}_{\mathcal{D}_0}[Y|Z = z] = \mathbb{E}_{\mathcal{D}_1}[Y|Z = z]$ with $z = \lambda(x)$.

Classification According to the definition, we have:

$$\mathcal{L}_0(h, \lambda) = \mathbb{E}_{\mathcal{D}_0} \log(1 + \exp(-yh(z)))$$

Since the optimal predictor on the logistic loss is the log-conditional density ratio: $h_0^\star(z) = \log\left(\frac{\mathcal{D}_0(Y=1|Z=z)}{\mathcal{D}_0(Y=-1|Z=z)}\right)$. Observe that in the binary classification with $Y = \{-1, 1\}$, we have $\mathcal{D}_0(Y = 1|Z = z) = \frac{1}{2}(1 + \mathbb{E}_{\mathcal{D}_0}[Y|Z = z])$ and $\mathcal{D}_0(Y = -1|Z = z) = \frac{1}{2}(1 - \mathbb{E}_{\mathcal{D}_0}[Y|Z = z])$, then we have:

$$h_0^\star(z) = \log\left(\frac{1 + \mathbb{E}_{\mathcal{D}_0}[Y|Z = z]}{1 - \mathbb{E}_{\mathcal{D}_0}[Y|Z = z]}\right)$$

As for $\mathcal{D}_1$, we adopt the same strategy and we have $\log\left(\frac{1+\mathbb{E}_{\mathcal{D}_0}[Y|Z=z]}{1-\mathbb{E}_{\mathcal{D}_0}[Y|Z=z]}\right) = \log\left(\frac{1+\mathbb{E}_{\mathcal{D}_1}[Y|Z=z]}{1-\mathbb{E}_{\mathcal{D}_1}[Y|Z=z]}\right)$, then we have $\mathbb{E}_{\mathcal{D}_0}[Y|Z = z] = \mathbb{E}_{\mathcal{D}_1}[Y|Z = z]$.

As for the predictive parity, since we have $\mathbb{E}_{\mathcal{D}_0}[Y|Z = z] = \mathbb{E}_{\mathcal{D}_1}[Y|Z = z]$ and $h^\star = h_1^\star = h_2^\star$, then we have $\mathbb{E}_{\mathcal{D}_0}[Y|h^\star(z)] = \mathbb{E}_{\mathcal{D}_1}[Y|h^\star(z)]$.

B APPROXIMATION ERROR

Theorem 2 (Approximation Error Gap)**.** *Suppose that (1)* ***Smooth Predictive Loss****. The first-order derivatives and second-order derivatives of $\mathcal{L}$ are Lipschitz continuous; (2)* ***Non-singular Hessian matrix****. We assume $\nabla_{h_0,h_0}\mathcal{L}_0(h_0, \lambda), \nabla_{h_1,h_1}\mathcal{L}_1(h_1, \lambda)$, the Hessian matrix of the inner optimization problem, are invertible. (3)* ***Bounded representation and predictor function****. We assume the λ and h are bounded, i.e., $\|\lambda\|, \|h\|$ are upper bounded by the predefined positive constants. Then the approximation error between the ground truth and algorithmic estimated gradient w.r.t. the representation is be upper bounded by:*

$$\|grad(\lambda) - \tilde{grad}^{\delta}(\lambda)\| = \mathcal{O}(\kappa\epsilon + \epsilon + \delta).$$

Proof. We denote grad(λ) as the ground truth gradient w.r.t. λ in outer-level loss (given the optimal predictor $h_0^\star$, $h_1^\star$). Then we aim to bound

$$\|\text{grad}(\lambda) - \tilde{\text{grad}}^{\delta}(\lambda)\|$$

We first introduce the following terms for facilitating the proof:

$A_0^\epsilon = \nabla_{h_0}\nabla_\lambda \mathcal{L}_0(h_0^\epsilon, \lambda), A_1^\epsilon = \nabla_\lambda \nabla_{h_1}\mathcal{L}_1(h_1^\epsilon, \lambda), A_0^\star = \nabla_\lambda \nabla_{h_0}\mathcal{L}_0(h_0^\star, \lambda), A_1^\star = \nabla_\lambda \nabla_{h_1}\mathcal{L}_1(h_1^\star, \lambda),$
$B_0^\epsilon = \nabla_\lambda \mathcal{L}_0(h_0^\epsilon, \lambda), B_1^\epsilon = \nabla_\lambda \mathcal{L}_1(h_1^\epsilon, \lambda), B_0^\star = \nabla_\lambda \mathcal{L}_0(h_0^\star, \lambda), B_1^\star = \nabla_\lambda \mathcal{L}_1(h_1^\star, \lambda),$

$$\mathbf{p}_0^\star = \left(\nabla_{h_0}^2 \mathcal{L}_0(h_0^\star, \lambda)\right)^{-1} \left(\nabla_{h_0}\mathcal{L}_0(h_0^\star, \lambda) + \kappa(h_0^\star - h_1^\star)\right),$$

$$\mathbf{p}_1^\star = \left(\nabla_{h_1}^2 \mathcal{L}_1(h_1^\star, \lambda)\right)^{-1} \left(\nabla_{h_1}\mathcal{L}_1(h_1^\star, \lambda) - \kappa(h_0^\star - h_1^\star)\right).$$

Then the approximation error gap can be expressed as:

$$\begin{aligned}\|\text{grad}(\lambda) - \tilde{\text{grad}}^{\delta}(\lambda)\| &= \| \left(B_0^\star - A_0^\star \mathbf{p}_0^\star + B_1^\star - A_1^\star \mathbf{p}_1^\star\right) - \left(B_0^\epsilon - A_0^\epsilon \mathbf{p}_0^\delta + B_1^\epsilon - A_1^\epsilon \mathbf{p}_1^\delta\right) \| \\ &\leq \sum_{i=0}^{1} \|B_i^\star - B_i^\epsilon\| + \sum_{i=0}^{1} \|A_i^\star \mathbf{p}_i^\star - A_i^\delta \mathbf{p}_i^\delta\|\end{aligned}$$

Due to the symmetric of $\mathcal{D}_0$ and $\mathcal{D}_1$, we only focus on the term on $i = 0$, the the upper bound in $i = 1$ can be derived analogously.

As for bounding $\|B_0^\star - B_0^\epsilon\|$, since we assume first order derivative of the loss is Lipschitz functions (with constant L_1), then we have :

$$\|B_0^\star - B_0^\epsilon\| \leq L_1 \|h_0^\star - h_0^\epsilon\| \leq \epsilon L_1$$

Then the second term can be upper bounded by three terms:

$$\|A_0^\star \mathbf{p}_0^\star - A_0^\delta \mathbf{p}_0^\delta\| \leq \underbrace{\|A_0^\star \mathbf{p}_0^\star - A_0^\star \mathbf{p}_0\|}_{(1)} + \underbrace{\|A_0^\star \mathbf{p}_0 - A_0^\epsilon \mathbf{p}_0\|}_{(2)} + \underbrace{\|A_0^\epsilon \mathbf{p}_0 - A_0^\epsilon \mathbf{p}_0^\delta\|}_{(3)}$$

Before estimating the upper bound, we first demonstrate $\|A_0^\epsilon\|$ and $\|A_0^\star\|$ are also bounded.

Since we assume λ and h are bounded (assuming the bounded constant as η and ϕ), the second order derivative are Lipschitz (with constant L_2). Then we consider another fixed point $(\lambda', h_0^\star(\lambda'))$ with bounded second order derivative: $A_0 = \nabla^2_{h_0,\lambda} \mathcal{L}_0(h_0^\star(\lambda'), \lambda')$ and $\|A_0\| \leq A$. We have:

$$\|A_0^\star - A_0\|_2 \leq L_2 \|[h_0^\star(\lambda), \lambda] - [h_0^\star(\lambda'), \lambda']\|_2 \leq L_2 \sqrt{\eta^2 + \phi^2}$$

Thus we have $\|A_0^\star\| \leq A + L_2\sqrt{\eta^2 + \phi^2} = A_{\text{sup}}^\star$. As for the second derivative at point h_0^ϵ, it can be upper bounded analogously with a similar constant A_{sup}^ϵ.

The upper bound of term (1) We have:

$$\|A_0^\star \mathbf{p}_0^\star - A_0^\star \mathbf{p}_0\| \leq \|A_0^\star\| \|\mathbf{p}_0^\star - \mathbf{p}_0\|$$

We have proved $\|A_0^\star\|$ is upper bounded by $A_{\text{sup}}^\star$. We additionally introduce the following auxiliary terms:

$$P_0^\star = \left(\nabla^2_{h_0} \mathcal{L}_0(h_0^\star, \lambda)\right)^{-1}, P_0^\epsilon = \left(\nabla^2_{h_1} \mathcal{L}_1(h_1^\star, \lambda)\right)^{-1}.$$
$$b_0^\star = \nabla_{h_0} \mathcal{L}_0(h_0^\star, \lambda) + \kappa(h_0^\star - h_1^\star), b_0^\epsilon = \nabla_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda) + \kappa(h_0^\epsilon - h_1^\epsilon)$$

Then we have:

$$\begin{aligned}
\|\mathbf{p}_0^\star - \mathbf{p}_0\| &= \|P_0^\star b_0^\star - P_0^\epsilon b_0^\epsilon\| \\
&\leq \|P_0^\star b_0^\star - P_0^\star b_0^\epsilon\| + \|P_0^\star b_0^\epsilon - P_0^\epsilon b_0^\epsilon\| \\
&\leq \|P_0^\star\| \|b_0^\star - b_0^\epsilon\| + \|b_0^\epsilon\| \|P_0^\star - P_0^\epsilon\|
\end{aligned}$$

As for the $\|P_0^\star\|$, since we assume the Hessian matrix is invertible thus its norm is upper bounded by some constant (denoted as A_{-1}). As for $\|b_0^\star - b_0^\epsilon\|$, we have:

$$\begin{aligned}
\|b_0^\star - b_0^\epsilon\| &\leq \|\nabla_{h_0} \mathcal{L}_0(h_0^\star, \lambda) - \nabla_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda)\| + 2\kappa\epsilon \\
&\leq \epsilon L_1 + 2\kappa\epsilon
\end{aligned}$$

Thus we have $\|P_0^\star\| \|b_0^\star - b_0^\epsilon\| \leq A_{-1}(\epsilon L_1 + 2\kappa\epsilon)$.

As for $\|b_0^\epsilon\|$, we can easily verify that it is indeed bounded by some constant b. For the first term, we can adopt the same strategy in proving bounded $\|A_0^\star\|$. As for the second term in b_0^ϵ, it is upper bounded by $2\kappa\phi$, due to the bounded predictor.

We now demonstrate $\|P_0^\star - P_0^\epsilon\|$. Denoting $\Delta = (P_0^\star)^{-1} - (P_0^\epsilon)^{-1}$, then according to the second order Lipschitz assumption, we have: $\|\Delta\| \leq \epsilon L_2$. Plugging in the result, we have:

$$\|P_0^\star - P_0^\epsilon\| = \|(P_0^\star)\Delta(P_0^\epsilon)\| \leq \|P_0^\star\| \|\Delta\| \|P_0^\epsilon\| \leq (A_{-1})^2 L_2 \epsilon$$

We still adopt the assumption that the bounded Hessian-inverse matrix by A_{-1}.

Plugging in all the results, we have:

$$(1) \leq A_1(\epsilon L_1 + 2\kappa\epsilon) + b(A_1)^2 L_2 \epsilon := \mathcal{O}(\kappa\epsilon + \epsilon)$$

The upper bound of term (2) We have:

$$\|A_0^\star \mathbf{p}_0 - A_0^\epsilon \mathbf{p}_0\| \leq \|\mathbf{p}_0\|_2 \|A_0^\star - A_0^\epsilon\|$$

Since we assume the loss is second-order Lipschitz, thus we have

$$\|A_0^\star - A_0^\epsilon\| = \|\nabla_\lambda \nabla_{h_0} \mathcal{L}_0(h_0^\star, \lambda) - \nabla_\lambda \nabla_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda)\| \leq L_2 \|h_0^\star - h_0^\epsilon\| \leq \epsilon L_2$$

We can also demonstrate $\|\mathbf{p}_0\|$ is bounded. According to the definition we have:

$$\begin{aligned}\|\mathbf{p}_0\| &\leq \| \left(\nabla_{h_0}^2 \mathcal{L}_0(h_0^\epsilon, \lambda)\right)^{-1} \| \| \left(\nabla_{h_0} \mathcal{L}_0(h_0^\epsilon, \lambda) + \kappa(h_0^\epsilon - h_1^\epsilon)\right) \| \\ &\overset{(i)}{\leq} A_{-1}(L_1 \|h_0^\star - h_0^\epsilon\|_2 + 2\kappa\phi) \\ &\overset{(ii)}{\leq} A_{-1}(\epsilon L_1 + 2\kappa\phi)\end{aligned}$$

For (i), we assume: 1) the Hessian matrix is invertible thus its norm is surely upper bounded by some constant (denoted as A_{-1}), 2) the first-order derivative is Lipschitz (bounded by L_1), 3) the predictor h is bounded. For (ii), we adopt the definition of h_0^ϵ.

Therefore, the upper bound for Term (2) is formulated as:

$$(2) \leq \epsilon L_2 A_{-1}(\epsilon L_1 + 2\kappa\phi) := \mathcal{O}(\kappa\epsilon)$$

The upper bound of term (3) We have:

$$\|A_0^\epsilon \mathbf{p}_0 - A_0^\epsilon \mathbf{p}_0^\delta\| \leq \|A_0^\epsilon\| \|\mathbf{p}_0 - \mathbf{p}_0^\delta\| \leq \delta A_{\text{sup}}^\epsilon = \mathcal{O}(\delta)$$

Through the upper bound in (1)-(3), we finally have the error between the estimated and ground-truth gradient:

$$\|\text{grad}(\lambda) - \tilde{\text{grad}}^\delta(\lambda)\| = \mathcal{O}(\kappa\epsilon + \epsilon + \delta)$$

□

C THE CONVERGENCE BEHAVIOR

For the sake of completeness, we provide the convergence analysis of the proposed algorithm.

Proposition 2. *We execute the implicit alignment algorithm (Algo. 1), obtaining a sequence of $\lambda_1, \ldots, \lambda_k, \ldots$. Supposing the fair constraint κ is fixed. The optimization tolerances are summable: $\sum_k \epsilon_k^2 \leq +\infty$ and $\sum_k \delta_k^2 \leq +\infty$, then λ_k is proved to be converged with*

$$\lim_{k \to \infty} \lambda_k = \lambda^\star.$$

If the stationary point $\lambda^\star$ is also within the bounded norm, then we have:

$$\textit{grad}(\lambda^\star) = 0.$$

Proof. We denote the entire outer-level loss w.r.t. λ as $\mathcal{L}(\lambda)$, by the assumption the β-smooth loss $\mathcal{L}$. Then at iteration $k+1$ and k, we have:

$$\begin{aligned}\mathcal{L}(\lambda_{k+1}) &\leq \mathcal{L}(\lambda_k) - \text{grad}(\lambda_k)^T(\lambda_k - \lambda_{k+1}) + \frac{\beta}{2}\|\lambda_{k+1} - \lambda_k\|^2 \\ &= \mathcal{L}(\lambda_k) - \left(\text{grad}(\lambda_k) - \tilde{\text{grad}}^\delta(\lambda_k) + \tilde{\text{grad}}^\delta(\lambda_k)\right)^T(\lambda_k - \lambda_{k+1}) + \frac{\beta}{2}\|\lambda_{k+1} - \lambda_k\|^2 \\ &= \mathcal{L}(\lambda_k) - \left(\text{grad}(\lambda_k) - \tilde{\text{grad}}^\delta(\lambda_k)\right)^T(\lambda_k - \lambda_{k+1}) - \tilde{\text{grad}}^\delta(\lambda_k)(\lambda_k - \lambda_{k+1}) + \frac{\beta}{2}\|\lambda_{k+1} - \lambda_k\|^2\end{aligned}$$

Since we assume the representation is within the bounded norm, the projection onto the convex set are non-expansive operators (Boyd et al., 2004). Then for any point p, q, we have $\|\text{proj}(p) - \text{proj}(q)\|^2 \leq (p-q)^T(\text{proj}(p) - \text{proj}(q))$. Then we set λ_k and $\lambda_{k+1} = \lambda_k - \frac{1}{\beta}\tilde{\text{grad}}^\delta(\lambda_k)$, we have:

$$\|\lambda_k - \lambda_{k+1}\|^2 \leq \frac{1}{\beta}(\tilde{\text{grad}}^\delta(\lambda_k))^T(\lambda_k - \lambda_{k+1})$$

Plugging into the results, we have:

$$\begin{aligned}\mathcal{L}(\lambda_{k+1}) &\leq \mathcal{L}(\lambda_k) - \left(\text{grad}(\lambda_k) - \tilde{\text{grad}}^{\delta}(\lambda_k)\right)^T (\lambda_k - \lambda_{k+1}) - \frac{\beta}{2}\|\lambda_{k+1} - \lambda_k\|^2 \\ &\leq \mathcal{L}(\lambda_k) + \|\text{grad}(\lambda_k) - \tilde{\text{grad}}^{\delta}(\lambda_k)\|\|\lambda_k - \lambda_{k+1}\| - \frac{\beta}{2}\|\lambda_{k+1} - \lambda_k\|^2\end{aligned}$$

Rearranging the inequality, we have:

$$\frac{\beta}{2}\|\lambda_{k+1} - \lambda_k\|^2 - \|\text{grad}(\lambda_k) - \tilde{\text{grad}}^{\delta}(\lambda_k)\|\|\lambda_k - \lambda_{k+1}\| + (\mathcal{L}(\lambda_{k+1}) - \mathcal{L}(\lambda_k)) \leq 0$$

Then we have:

$$\|\lambda_{k+1} - \lambda_k\| \leq \frac{1}{\beta}\left(\|\text{grad}(\lambda_k) - \tilde{\text{grad}}^{\delta}(\lambda_k)\| + \sqrt{\|\text{grad}(\lambda_k) - \tilde{\text{grad}}^{\delta}(\lambda_k)\|^2 - 2\beta\left(\mathcal{L}(\lambda_{k+1}) - \mathcal{L}(\lambda_k)\right)}\right)$$

By denoting $B_k = \|\text{grad}(\lambda_k) - \tilde{\text{grad}}^{\delta}(\lambda_k)\|$ and $C_k = \mathcal{L}(\lambda_{k+1}) - \mathcal{L}(\lambda_k)$. Then we have:

$$\begin{aligned}\|\lambda_{k+1} - \lambda_k\|^2 &\leq \frac{1}{\beta^2}\left(B_k^2 + B_k^2 - 2\beta C_k + 2B_k\sqrt{B_k^2 - 2\beta C_k}\right) \\ &\leq \frac{1}{\beta^2}\left(B_k^2 + B_k^2 - 2\beta C_k + B_k^2 + B_k^2 - 2\beta C_k\right) \\ &= \frac{4}{\beta^2}[\|\text{grad}(\lambda_k) - \tilde{\text{grad}}^{\delta}(\lambda_k)\|_2^2 - 2\beta\left(\mathcal{L}(\lambda_{k+1}) - \mathcal{L}(\lambda_k)\right)]\end{aligned}$$

Taking sum over k, we have:

$$\begin{aligned}\sum_{k=1}^{+\infty}\|\lambda_{k+1} - \lambda_k\|^2 &\leq \frac{4}{\beta^2}\sum_{k=1}^{+\infty}\|\text{grad}(\lambda_k) - \tilde{\text{grad}}^{\delta}(\lambda_k)\|_2^2 - \frac{8}{\beta}(\lim_{k\to\infty}\mathcal{L}(\lambda_{k+1}) - \mathcal{L}(\lambda_1)) \\ &\leq \frac{4}{\beta^2}\sum_k[(C+\kappa)^2\epsilon_k^2 + \delta_k^2] - \frac{8}{\beta}\left(\lim_{k\to\infty}\mathcal{L}(\lambda_{k+1}) - \mathcal{L}(\lambda_1)\right) < +\infty\end{aligned}$$

Since 1) the first term on the right side is finite, because the optimization tolerance is summable; 2) the second term is also finite, because the loss is assumed to be bounded. Then the upper bound is finite. In order to satisfy this condition, on the left side we should have:

$$\lim_{k\to\infty}\lambda_{k+1} - \lambda_k = 0$$

By adopting the definition $\lambda_{k+1} = \text{Proj}(\lambda_k - \tilde{\text{grad}}^{\delta}(\lambda_k))$ and $\lim_{k\to\infty}\tilde{\text{grad}}^{\delta}(\lambda_k) = \text{grad}(\lambda_k)$ (Based on theorem 1, the limit of the optimization tolerance is zero), then we have:

$$\lambda^\star = \text{proj}(\lambda^\star - \text{grad}(\lambda^\star))$$

Where $\lambda^\star = \lim_{k\to+\infty}\lambda_{k+1} = \lim_{k\to+\infty}\lambda_k$. Since the projection is on the bounded norm L_{norm} and $\lambda^\star$ is within the bounded norm space, thus if $\lambda^\star - \text{grad}(\lambda^\star)$ is within the bounded norm space, we have:

$$\text{grad}(\lambda^\star) = 0$$

Else if $\lambda^\star - \text{grad}(\lambda^\star)$ is outside the bounded norm space, then according to the definition, the projection of $\lambda^\star - \text{grad}(\lambda^\star)$ is surely on the *boundary* of the L_{norm} space, with $\|\text{proj}(\lambda^\star - \text{grad}(\lambda^\star))\| = L_{\text{norm}}$. However, we have assumed the $\lambda^\star$ is *within* the bounded norm space with $\|\lambda^\star\| < L_{\text{norm}}$, which leads to the contradiction. Based on these discussions, we finally have:

$$\text{grad}(\lambda^\star) = 0$$

□

D Additional Experimental details and Results

D.1 Additional Details

Toxic Comments We split the training, validation and testing set as 70%, 10% and 20%. We adopt Adam optimizer with learning rate 10^{-3} and eps 10^{-3}. The batch-size is set as 500 for each subgroup and we use sampling with replacement to run the explicit algorithm with maximum epoch 100. The fair coefficient is generally set as $\kappa = 0.1 \sim 0.001$. As for the inner-optimization step, the iteration number is 20 and the iteration in running conjugate gradient approach is 10.

CelebA The training/validation/test set are around 82K, 18K and 18K. We also adopt the Adam optimizer with learning rate on $\lambda : 10^{-5} \sim 10^{-4}$ and $h : 10^{-3}$. The batch-size is set as 64 for each subgroup and we iterate the whole dataset as one epoch. The maximum running epoch is set as 20 and the iteration in running conjugate gradient approach is 10.

Law We split the training, validation and testing set as 70%, 10% and 20%. Then we adopt Adam optimizer with learning rate 10^{-3} and eps 10^{-3}. The batch-size is set as 500 for each subgroup and we use sampling with replacement to run the implicit algorithm, with the maximum epoch 100. We adopt the MSE loss in the regression. The fair coefficient is generally set as $\kappa = 0.1 \sim 10^{-4}$. As for the inner-optimization, the iteration number is 20 and the iteration in running conjugate gradient is 10. In computing the sufficiency gap in the regression, we sample 33 points to compute the gap.

NLSY We split the training, validation and testing set as 70%, 10% and 20%. Then we adopt Adam optimizer with learning rate 10^{-3} and eps 10^{-3}. The batch-size is set as 500 for each subgroup and we use sampling with replacement to run the implicit algorithm, with maximum epoch 100. We adopt the MSE loss in the regression. The fair coefficient is generally set as $\kappa = 0.1 \sim 10^{-4}$. As for the inner-optimization, the iteration number is 20 and the iteration in running conjugate gradient is 10. In computing the sufficiency gap, we sample 33 points to compute the sufficiency gap.

D.2 Additional Empirical results

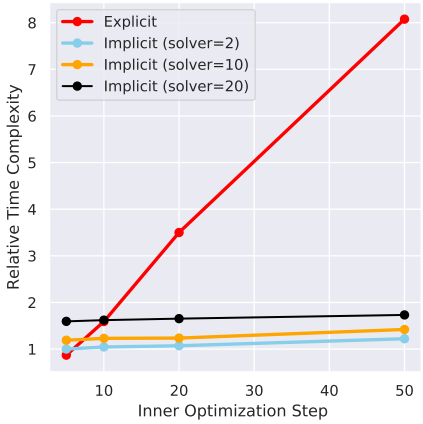


Figure 9: Computational time between T-step explicit and implicit approach in CelebA. Specifically, solver $= 2$ indicates the the conjugate gradient is executed 2 iterations. The results reveals the benefits of implicit approach: avoiding the back-propagation through the inner-optimization path. In contrast, the time complexity in explicit approach linearly increases with the inner-optimization step, which is consistent with our analysis.

Computational complexity To show the efficiency of the implicit approach, we empirically compare the computational complexity of the T-step explicit alignment and implicit approach (for different iterations of conjugate gradient solver.) The experimental results verified the efficiency of the implicit approach, where a significant large inner-optimization step does not considerably increase the computational time.

Gradient evolution We also visualize the gradient norm of the representation λ in the Toxic dataset, shown in Fig. 10. The results verify the convergence behavior and the gradient norm finally tends to zero.

D.3 Discussion with non-deep learning baselines

In order to show the effectiveness of the proposed approach, we additionally compare the FAHT (Zhang & Ntoutsi, 2019), a decision tree based fair classification approach. We evaluated the empirical performance on Toxic comments dataset.

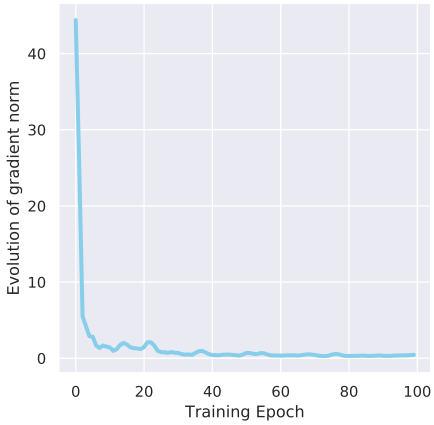


Figure 10: Gradient Norm evolution w.r.t. representation λ in Toxic comments dataset. We visualize the norm of $\tilde{\text{grad}}^{\delta}(\lambda)$ at each training epoch, which suggests a convergence behavior and the gradient finally tends to zero.

Table 5: Comparison with Fairness Aware Decision Tree

Method	Accuracy (↑)	ΔSuf_C (↓)
FAHT	0.596	0.397
Implicit	0.760	0.051

The implicit approach demonstrates the considerable better results, which may come from two aspects: (1) the Toxic task is a high-dimensional classification problem ($x \in \mathbb{R}^{748}$), where the deep learning based approach is more effective in handling the high-dim dataset. (2) The FAHT aims to realize the statistical parity (the independence rule), which is *not compatible* with the sufficiency. According to the analysis of (Barocas et al., 2019), when the sensitive attribute (A) and label (Y) are not independent (This has been justified by computing their Pearson Correlation coefficient), the sufficiency and independence cannot both hold.

D.4 SUFFICIENCY GAP IN REGRESSION

We visualize the sufficiency gap of NLSY dataset.

E COMPLEMENTARY TECHNICAL DETAILS

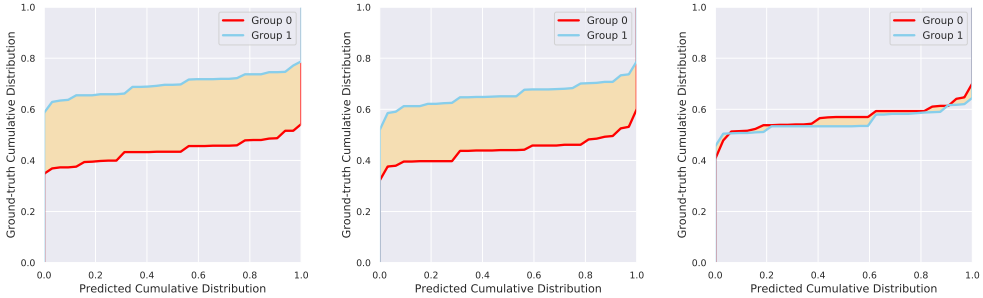
We present complementary details that are related to the paper.

E.1 CONJUGATE GRADIENT METHOD

We present the Conjugate Gradient (CG) algorithm in Algo. 2 through `autograd`. In the conventional CG algorithm with objective $\frac{1}{2}x^T AX - bX$, we need to estimate AX and compute its residual and update X. Since in our problem setting, the $A = \nabla^2_{h_0}\mathcal{L}_0(h_0^\epsilon, \lambda)$, then computing AX can be realized through Hessian-vector product through `autograd`, denoted as function F in the paper. i.e., $\nabla^2_{h_0}\mathcal{L}_0(h_0^\epsilon, \lambda)X = F(x)$.

Below we provided a simple PyTorch code for realizing the Hessian Vector product.

```
import torch
def hessain_vector_product(loss,model,vector):
    # loss: the defined loss
    # model: the model in computing the Hessian
    # vector: the required vector in computing Hessian-vector product
```



(a) ERM (b) Fair Mix-up (c) Implicit

Figure 11: Illustration of the *sufficiency gap* in NLSY dataset. The ERM and mix-up suffer the high predictive sufficiency-gap, while the proposed implicit alignment can significantly mitigate the sufficiency gap. In contrast, the probability calibration is not improved. This results also verifies the inequivalence between the sufficiency gap and calibration gap (Liu et al., 2019).

Algorithm 2 Conjugate Gradient Method

Ensure: Function F that computes Hessian-vector product through `autograd`, initial value X_0, bias vector B.

Computing Residual: $r_0 = B - F(X_0)$
Set $p_0 = r_0$
for inner_iterations k **do**
  Computing $\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T F(p_k)}$
  $X_{k+1} \leftarrow X_k + \alpha_k p_k$
  $r_{k+1} \leftarrow r_k - \alpha_k F(p_k)$
  If r_{k+1} is sufficiency small, then stop.
  $\beta_k \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
  $p_{k+1} \leftarrow r_{k+1} + \beta_k p_k$
end for
return X_{k+1}

```
    partial_grad = torch.autograd.grad(loss, model_parameters(),
    create_graph=True)
    flat_grad = torch.cat([g.contiguous().view(-1) for g in partial_grad
    ])
    h = torch.sum(flat_grad * vector_to_optimize)
    hvp = torch.autograd.grad(h, model.parameters())
    return hvp
```

Listing 1: Simple demo in computing Hessian vector product

E.2 Calibration Gap in the Regression

Based on Kuleshov et al. (2018), we first compute the predicted cumulative distribution ($\hat{Y}_0$) of at point t: $D_0(\hat{Y}_0 \leq t) = \alpha$, then we compute the corresponding ground truth cumulative distribution (Y_0) at point t. By changing t, we obtain several points on function $\mathcal{D}_0(Y \leq t|\hat{Y}_0 \leq t) = \beta$. Then the regression is probabilistic calibrated when $\alpha \equiv \beta$. From this perspective, the zero calibration gap can guarantee a zero sufficiency gap. But the inverse is not necessarily true, as our experimental results suggest, a small sufficiency gap can lead to either small or large calibration gap. Thus it can be quite promising to explore their inherent relations and trade-off in the fair regression.