OUKA
The University of Osaka
Institutional Knowledge Archive

Title	Predictive Fair Representation Learning with Variational Autoencoders
Author(s)	Yamada, Tatsuya; Konishi, Takuya; Kawahara, Yoshinobu
Citation	New Generation Computing. 2026, 44(16)
Version Type	VoR
URL	https://hdl.handle.net/11094/104796
rights	This article is licensed under a Creative Commons Attribution 4.0 International License.
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

New Generation Computing (2026) 44:16
https://doi.org/10.1007/s00354-026-00320-0

RESEARCH PAPER

Predictive Fair Representation Learning with Variational Autoencoders

Tatsuya Yamada[1] · **Takuya Konishi**[1,2] · **Yoshinobu Kawahara**[1,2]

Received: 5 June 2025 / Accepted: 10 March 2026
© The Author(s) 2026

Abstract
Fairness is a critical concern in machine learning, as biases in prediction outcomes with respect to sensitive attributes such as gender and race have raised ethical and societal issues in real-world applications. To address this, fair representation learning aims to extract essential information from data as representations that are useful but independent of sensitive attributes. However, existing methods often suffer from limited predictive performance when their learned representations are applied to downstream tasks. In this study, we propose predictive FairDisCo (PdFairDisCo), an extension of FairDisCo to learn fair representations with variational autoencoders. PdFairDisCo enhances predictive performance by incorporating contrastive losses into the objective function. In addition, we introduce two oversampling methods for PdFairDisCo to mitigate bias by balancing the proportion of the sensitive attribute in training data. We demonstrate the effectiveness of PdFairDisCo through experiments. The experiments also show that the oversampling methods can further improve the performance of fairness.

Keywords Fair representation learning · Variational autoencoders · Contrastive loss · Oversampling

✉ Yoshinobu Kawahara
kawahara@ist.osaka-u.ac.jp

Tatsuya Yamada
t-yamada@ist.osaka-u.ac.jp

Takuya Konishi
konishi@ist.osaka-u.ac.jp

[1] Graduate School of Information Science and Technology, The University of Osaka, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

[2] Center for Advanced Intelligence Project, RIKEN, 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan

Published online: 25 March 2026

1 Introduction

Artificial intelligence (AI) has become an essential part of our lives in recent years. AI algorithms are used in many applications and are increasingly expected to play a greater role in various fields. When deploying AI systems in real-world scenarios, a significant challenge is to ensure fair algorithms [1]. For instance, it is reported that COMPAS, the software for decision making in court, has overestimated the probability of an African American committing a crime again [2]. In another case, advertisements promoting job opportunities in science, technology, engineering, and math, were not sufficiently displayed to women [3]. These biased outcomes from unfair algorithms can be especially serious in areas with critical decision making such as politics, healthcare, or finance.

Various studies have been devoted to achieving fair AI algorithms. In the machine learning literature, unfair algorithms often arise in learning from biased training data, and many studies explore approaches to eliminate biases in sensitive attributes such as gender and race. One promising direction is the study of fair representation learning, which aims at removing biases from representations of data [4–6]. In addition, the proportion of sensitive attributes is sometimes imbalanced in training data, which can be another source of biases for learning algorithms. Several studies have proposed to increase training data with fewer sensitive attributes [7].

In this study, we focus on FairDisCo [5], a method for fair representation learning with variational autoencoders (VAEs) [8]. This recently proposed approach uses distance covariance to learn fair representations such that they are no longer associated with sensitive attributes. While FairDisCo provides task-agnostic and general-purpose representations, learned representations may achieve suboptimal predictive performance in tasks of interest. Moreover, FairDisCo can still face the imbalanced problem of sensitive attributes. Even if FairDisCo attains nearly fair representations, small biases could be amplified if training data are imbalanced in downstream tasks.

To address these issues, we first propose predictive FairDisCo (PdFairDisCo). While FairDisCo remains an appropriate choice for task-agnostic representations, PdFairDisCo is designed for learning fair and task-relevant representations when class label information for a task of interest is available. The key idea is to exploit contrastive losses [9, 10] such as a triplet loss [11–13] and a supervised contrastive loss (SCL) [14] to integrate the label information into representations. We also utilize oversampling for the imbalanced problem of sensitive attributes. Oversampling balances training data by increasing the number of samples with a minority sensitive attribute value. We incorporate oversampling for PdFairDisCo and propose two methods. We experimentally demonstrate the effectiveness of our approaches. The results indicate that PdFairDisCo consistently improves predictive performance and, in several cases, also yields fairer representations. Furthermore, our oversampling methods provide additional gains in fairness.

The remainder of the paper is organized as follows. Section 2 provides background information, and Sect. 3 describes the proposed methods, PdFairDisCo, and oversampling methods. Section 4 presents the experimental results to evaluate our methods. In Sect. 5, we review other fair representation learning methods and oversampling methods, and finally Sect. 6 concludes the study.

2 Background

In this section, we review FairDisCo, which is the basis of our approach. FairDisCo builds on the VAE framework by adding a fairness constraint with distance covariance. We first overview the notation and problem setup and then describe the formulation of FairDisCo.

2.1 Preliminaries

Suppose that we have paired data $(\boldsymbol{x}, s)$, where $\boldsymbol{x} \in \mathcal{X}$ is a set of non-sensitive features for a subject, and $s \in \mathcal{S}$ represents the associated sensitive attribute (e.g., gender and race). In the following, we suppose that $\boldsymbol{x}$ takes a value in the subset of the d_x-dimensional Euclidean space, i.e., $\mathcal{X} \subseteq \mathbb{R}^{d_x}$, but our formulation below can hold if $\boldsymbol{x}$ takes other discrete or mixed values. Since the number of possible values of a sensitive attribute is often finite (or grouped by several categories), we define s as a categorical variable with K attribute values, i.e., $\mathcal{S} = \{0, \ldots, K-1\}$.

Fair representation learning considers a transformation from $(\boldsymbol{x}, s)$ to a representation $\boldsymbol{z} \in \mathcal{Z}$, where we assume $\mathcal{Z} = \mathbb{R}^{d_z}$. This transformation is designed with two purposes: 1) to extract an informative representation from the original feature $\boldsymbol{x}$ and 2) to remove information about the sensitive attribute s for fairness. Once such a transformation is obtained, it can be applied to downstream tasks requiring fair prediction for $(\boldsymbol{x}, s)$. Note that ignoring s cannot be a solution for the second purpose. Even if s is excluded from the transformation, $\boldsymbol{x}$ can correlate with s, and the resulting $\boldsymbol{z}$ is still biased against s.

To promote the fairness of a representation $\boldsymbol{z}$, one can consider ensuring that $\boldsymbol{z}$ is statistically independent of a sensitive attribute s. A theoretically feasible approach to encourage this independence is to minimize mutual information defined by probability distributions for $\boldsymbol{z}$ and s. However, directly minimizing mutual information is generally intractable in practice. Several studies have proposed surrogate measures as an approximation or a lower bound on mutual information [4, 15, 16].

2.2 FairDisCo

Previous work has explored the use of VAEs for fair representation learning. Variational fair autoencoders (VFAEs) [17] and other VAE-based methods [4, 15, 16] have focused on extracting informative and fair representations. Building on this line of research, FairDisCo [5] introduces distance covariance to further enhance fairness in the learned representations and has demonstrated effectiveness over prior approaches.

FairDisCo constructs a transformation from paired data $(\boldsymbol{x}, s)$ to a representation $\boldsymbol{z}$ with a probabilistic model where $\boldsymbol{z}$ serves as a latent variable. We first define a conditional probability distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z}, s)$, typically implemented by a neural network called a decoder, where θ denotes the decoder's parameters. We also prescribe a prior distribution of $\boldsymbol{z}$ as multivariate Gaussian $p(\boldsymbol{z}) := \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ with zero mean and identity covariance. Applying Bayes' rule, we can obtain the corresponding conditional posterior distribution $p(\boldsymbol{z}|\boldsymbol{x}, s)$. As this true posterior is intractable due to the complexity

of marginalizing over z, we introduce a variational approximation $q_\phi(z|\boldsymbol{x}, s)$ implemented by another neural network called an encoder with parameters ϕ. Hereafter, we also refer to $p_\theta(\boldsymbol{x}|z, s)$ and $q_\phi(z|\boldsymbol{x}, s)$ as the decoder and encoder, respectively. In particular, we assume that the encoder $q_\phi(z|\boldsymbol{x}, s)$ is represented as the following Gaussian distribution as in standard VAEs:

$$q_\phi(z|\boldsymbol{x}, s) := \mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{x}, s), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}, s))) \tag{1}$$

where $\boldsymbol{\mu}_\phi(\boldsymbol{x}, s) \in \mathbb{R}^{d_z}$ is the mean vector parameterized by a neural network that takes $(\boldsymbol{x}, s)$ as input, and $\mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}, s))$ is the $d_z \times d_z$ diagonal covariance matrix whose diagonal elements consist of a vector $\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}, s) \in \mathbb{R}^{d_z}$. As in the mean vector, $\boldsymbol{\sigma}^2_\phi(\boldsymbol{x}, s)$ is also modeled by a neural network. Finally, FairDisCo performs a non-deterministic transformation to obtain z given $(\boldsymbol{x}, s)$ by sampling from the encoder: $z \sim q_\phi(z|\boldsymbol{x}, s)$.

To train the above probabilistic model, FairDisCo uses an objective function similar to that of VAEs. First, a reconstruction loss is defined as the negative expected log-likelihood of the decoder $p_\theta(\boldsymbol{x}|z, s)$:

$$\mathcal{L}_{\mathrm{re}} := -\mathbb{E}_{q_\phi(z|\boldsymbol{x},s)} \left[\ln p_\theta(\boldsymbol{x}|z, s)\right]. \tag{2}$$

In addition, a regularization term is employed to penalize the encoder $q_\phi(z|\boldsymbol{x}, s)$:

$$\mathcal{L}_{\mathrm{KL}} := D_{\mathrm{KL}}(q_\phi(z|\boldsymbol{x}, s) \parallel p(z)), \tag{3}$$

where $D_{\mathrm{KL}}(q \parallel p)$ is the Kullback–Leibler (KL) divergence for the probability distributions q and p. The sum of these two terms corresponds to the negative evidence lower bound for the probabilistic model, and minimizing it encourages the model to learn representations that better explain the observed data.

FairDisCo also adopts a penalty to encourage fairness in representations. As we explained in Sect. 2.1, mutual information can be computationally expensive to measure the independence between z and s. Instead, FairDisCo imposes a fairness penalty based on distance covariance [18] as a surrogate for mutual information. Specifically, let us consider some joint distribution of z and s, denoted as $p(z, s)$, and marginal distributions $p(z)$ and $p(s)$. Then, a distance covariance loss is given as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{dc}} &:= \sum_{s\in\mathcal{S}} \int_{\mathcal{Z}} |p(z, s) - p(z)p(s)|^2 \,\mathrm{d}z \\ &= \sum_{s\in\mathcal{S}} p(s)^2 \tilde{p}(s)^2 \int_{\mathcal{Z}} |p(z|s) - \tilde{p}(z|s)|^2 \,\mathrm{d}z,\end{aligned} \tag{4}$$

where

$$\tilde{p}(s) := \sum_{s'\in S\setminus\{s\}} p(s'), \quad \tilde{p}(z, s) := \sum_{s'\in S\setminus\{s\}} p(z, s'), \quad \tilde{p}(z|s) := \frac{\tilde{p}(z, s)}{\tilde{p}(s)}.$$

Here we denote $p(s')$ and $p(\boldsymbol{z}, s')$ the same distributions as $p(s)$ and $p(\boldsymbol{z}, s)$, respectively. In FairDisCo, $p(s)$ and $\tilde{p}(s)$ are estimated as empirical distributions of training data. The integral of $\boldsymbol{z}$ can be computed as a closed form by approximating $p(\boldsymbol{z}|s)$ and $\tilde{p}(\boldsymbol{z}|s)$ with the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x}, s)$. See Section 3.2 of [5] for more detailed computation of Eq. (4).

The final objective of FairDisCo is defined as follows:

$$\mathcal{L}_{\text{FairDisCo}} = \mathcal{L}_{\text{re}} + \mathcal{L}_{\text{KL}} + \beta \mathcal{L}_{\text{dc}}, \tag{5}$$

where $\beta \geq 0$ is a hyperparameter to control the trade-off between the distance covariance loss and the other two losses. Given a training dataset of $(\boldsymbol{x}, s)$, this objective function can be minimized with the mini-batch stochastic gradient descent method. In practice, each term can be computed approximately or exactly for each mini-batch of the dataset.

3 Proposed Method

In this section, we present PdFairDisCo and the oversampling methods that build on it. Both approaches share a common goal of improving the performance of downstream tasks after representation learning. We first describe the motivation behind these approaches and then formulate them.

3.1 PdFairDisCo

We consider a two-stage setting in which we first learn representations and then train a model for a downstream task using the learned representations. As illustrated in Sect. 2, FairDisCo learns a representation $\boldsymbol{z}$ that preserves information about features $\boldsymbol{x}$ and considers fairness regarding a sensitive attribute s. This representation is general-purpose and applicable irrespective of the subsequent task, which are often formulated as supervised learning problems following representation learning. However, such task-agnostic approach could offer suboptimal predictive performance even when target tasks are already evident at representation learning. In this study, we focus on classification problems as the downstream task and develop PdFairDisCo that leverages label information when it is available during representation learning. Although the resulting representations are no longer task-agnostic, PdFairDisCo makes the representations more task-relevant and predictive while still maintaining fairness.

Although we consider a two-stage setting, a natural alternative in task-specific scenarios is an end-to-end approach that directly optimizes the target task. While end-to-end deep learning has become common in recent years, additional data are often obtained after the initial training in practice. In such situations, it can be preferable to update the model by leveraging only the newly obtained data, for example to save computational resources. Under this setting, a two-stage framework like ours can still be a practical option for incorporating the new data, which we view as the downstream task. Although end-to-end models can also be retrained on the additional data, our

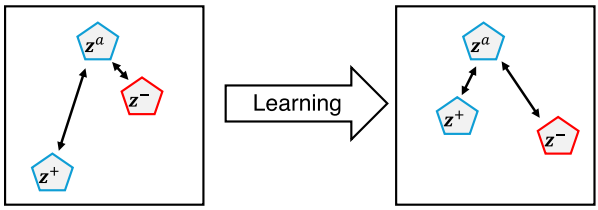


Fig. 1 A conceptual diagram of the effect of the triplet loss

approach offers several features compared to end-to-end frameworks. We provide further discussion in Sect. A.2.1.

Suppose that a pair of $(\boldsymbol{x}, s)$ in Sect. 2.1 is annotated to form a tuple $(\boldsymbol{x}, s, y)$, where $y \in \mathcal{Y}$ is a categorical label with L classes, i.e., $\mathcal{Y} = \{0, \ldots, L-1\}$. This label y is associated with a downstream classification task, and we aim to learn a better representation $\boldsymbol{z}$ such that y is more predictable.

Our proposed approach enhances FairDisCo by introducing an additional loss term that improves the predictive performance of the representations. In particular, we employ contrastive losses [9, 10], commonly used in deep metric learning, where the typical objective is to learn representations under supervision of distances between data. These losses typically use label information as explicit supervision, and we integrate them into FairDisCo so that $\boldsymbol{z}$ is encouraged to become closer to representations with the same label y and farther from those with different labels. As a result, we expect the learned representations to be more predictive in downstream classification tasks.

Formally, we consider the following form of contrastive losses. We first assume that we have a tuple $(\boldsymbol{x}^a, s^a, y^a)$, which we refer to as an anchor. Next, we assume that we have a positive tuple $(\boldsymbol{x}^+, s^+, y^+)$ such that $y^a = y^+$ and a negative tuple $(\boldsymbol{x}^-, s^-, y^-)$ where $y^a \neq y^-$. The corresponding representations are obtained from the encoder:

$$\boldsymbol{z}^a \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}^a, s^a), \quad \boldsymbol{z}^+ \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}^+, s^+), \quad \boldsymbol{z}^- \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}^-, s^-).$$

We further define the finite sets of positive and negative representations as a positive set $\mathcal{Z}^+$ and a negative set $\mathcal{Z}^-$, i.e., $\boldsymbol{z}^+ \in \mathcal{Z}^+$ and $\boldsymbol{z}^- \in \mathcal{Z}^-$, respectively. Then, contrastive losses are defined as the following expectation with respect to these representations:

$$\mathcal{L}_{\text{contrastive}} := \mathbb{E}_{\boldsymbol{z}^a, \mathcal{Z}^+, \mathcal{Z}^-}\left[\ell(\boldsymbol{z}^a, \mathcal{Z}^+, \mathcal{Z}^-)\right], \tag{6}$$

where $\ell(\boldsymbol{z}^a, \mathcal{Z}^+, \mathcal{Z}^-)$ represents an individual loss term for the samples $\boldsymbol{z}^a$, $\mathcal{Z}^+$, and $\mathcal{Z}^-$.

The form of a contrastive loss is determined by $\ell(\boldsymbol{z}^a, \mathcal{Z}^+, \mathcal{Z}^-)$, and we instantiate two specific losses. The first one is the SCL [14]:

$$\ell_\tau^{\text{SCL}}(\boldsymbol{z}^a, \mathcal{Z}^+, \mathcal{Z}^-) := -\frac{1}{|\mathcal{Z}^+|} \sum_{\boldsymbol{z}^+ \in \mathcal{Z}^+} \log \frac{\exp(\text{sim}(\boldsymbol{z}^a, \boldsymbol{z}^+)/\tau)}{\sum_{\boldsymbol{z} \in \mathcal{Z}^+ \cup \mathcal{Z}^-} \exp(\text{sim}(\boldsymbol{z}^a, \boldsymbol{z})/\tau)}, \tag{7}$$

where $\mathrm{sim}(z^a, z^+)$ denotes a function measuring the similarity between two representations z^a and z^+, e.g., cosine similarity, and $\tau > 0$ is a temperature parameter. This loss encourages the anchor representation z^a to move closer to all representations in $\mathcal{Z}^+$ and farther away from all representations in $\mathcal{Z}^-$.

Another loss we focus on is the triplet loss [11–13]. This loss is defined when the positive and negative sets consist of only one representation, i.e., $\mathcal{Z}^+ = \{z^+\}$ and $\mathcal{Z}^- = \{z^-\}$, respectively, as follows:

$$\ell_{p,m}^{\mathrm{triplet}}(z^a, z^+, z^-) := \max\left(\|z^a - z^+\|_p - \|z^a - z^-\|_p + m,\ 0\right), \tag{8}$$

where $\|\cdot\|_p$ denotes the L_p norm with $p \geq 1$, and $m > 0$ is a margin hyperparameter. This loss imposes a penalty when the distance between z^a and z^+ exceeds that between z^a and z^- by more than the margin m. Minimizing the triplet loss encourages z^a to be closer to z^+ than to z^-, as illustrated in Fig. 1. By substituting Eq. (7) or Eq. (8) into ℓ in Eq. (6), we obtain the corresponding contrastive loss.

The final objective of PdFairDisCo is defined by adding a contrastive loss $\mathcal{L}_{\mathrm{contrastive}}$ to the objective of FairDisCo in Eq. (5), as follows:

$$\mathcal{L}_{\mathrm{PdFairDisCo}} := \mathcal{L}_{\mathrm{re}} + \mathcal{L}_{\mathrm{KL}} + \beta\mathcal{L}_{\mathrm{dc}} + \alpha\mathcal{L}_{\mathrm{contrastive}}, \tag{9}$$

where $\alpha \geq 0$ is a hyperparameter that controls the contribution of the contrastive loss. In practice, PdFairDisCo is trained using the mini-batch gradient descent method in the same manner as FairDisCo. The detailed training procedure is shown in Algorithm 1. Given a training dataset $\mathcal{D} = \{(\boldsymbol{x}_n, s_n, y_n)\}_{n=1}^N$ of size N, we choose from $\mathcal{D}$ a mini-batch $\mathcal{D}_b = \{(\boldsymbol{x}_i, s_i, y_i)\}_{i=1}^B$ of size B ($B < N$). Next, we randomly sample representations $\boldsymbol{z}_i$ ($i = 1, \ldots, B$) from the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x}_i, s_i)$ and then compute each loss term in Eq. (9). After performing iterative gradient updates, we obtain the learned parameters ϕ of the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x}, s)$.

If we use an SCL on a mini-batch, the loss is computed using Algorithm 2. Given a mini-batch of representations and labels $\mathcal{D}_{\mathrm{contrastive}} = \{(\boldsymbol{z}_i, y_i)\}_{i=1}^B$, we first select an anchor representation z^a and classify the remaining representations into a positive set $\mathcal{Z}^+$ and a negative set $\mathcal{Z}^-$. Repeating the above procedure, we finally obtain the average SCL loss of the mini-batch.

Similarly, a triplet loss for a mini-batch is computed using Algorithm 3. After selecting z^a, $\mathcal{Z}^+$, and $\mathcal{Z}^-$, we sample a positive representation z^+ and a negative representation z^- uniformly from $\mathcal{Z}^+$ and $\mathcal{Z}^-$, respectively, and then compute a triplet loss value. In this way, computing a triplet loss requires selecting positive and negative representations. Although we adopt uniform random sampling to select positive and negative representations in Algorithm 3, we can also utilize more sophisticated sampling strategies [19]. Note that the SCL does not require any sampling strategies because it uses all positive and negative representations at the same time. In the following experiments, we mainly use the triplet loss and Algorithm 3 to evaluate PdFairDisCo.

Algorithm 1 Training algorithm of PdFairDisCo

Require: A dataset $\mathcal{D} = \{(\boldsymbol{x}_n, s_n, y_n)\}_{n=1}^N$, hyperparameters α and β.
Ensure: Encoder parameters ϕ.
Initialize ϕ and θ.
for $\mathcal{D}_b = \{(\boldsymbol{x}_i, s_i, y_i)\}_{i=1}^B \subseteq \mathcal{D}$ **do**
  for $(\boldsymbol{x}_i, s_i, y_i) \in \mathcal{D}_b$ **do**
    Sample $\boldsymbol{z}_i$ from $q_\phi(\boldsymbol{z}|\boldsymbol{x}_i, s_i)$.
  end for
  Compute $\mathcal{L}_{\text{re}}$ from $\{(\boldsymbol{x}_i, \boldsymbol{z}_i, s_i)\}_{i=1}^B$.
  Compute $\mathcal{L}_{\text{KL}}$ from $\{(\boldsymbol{x}_i, s_i)\}_{i=1}^B$.
  Compute $\mathcal{L}_{\text{dc}}$ from $\{(\boldsymbol{x}_i, s_i)\}_{i=1}^B$.
  Compute $\mathcal{L}_{\text{contrastive}}$ from $\{(\boldsymbol{z}_i, y_i)\}_{i=1}^B$.
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{re}} + \mathcal{L}_{\text{KL}} + \beta\mathcal{L}_{\text{dc}} + \alpha\mathcal{L}_{\text{contrastive}}$.
  Update parameters ϕ and θ by gradient descent for $\mathcal{L}$.
end for

Algorithm 2 Computing an SCL for mini-batch data

Require: A mini-batch $\mathcal{D}_{\text{contrastive}} = \{(\boldsymbol{z}_i, y_i)\}_{i=1}^B$, a hyperparameter τ.
Ensure: A contrastive loss $\mathcal{L}_{\text{contrastive}}$.
$\mathcal{L}_{\text{contrastive}} \leftarrow 0$.
for $(\boldsymbol{z}_i, y_i) \in \mathcal{D}_{\text{contrastive}}$ **do**
  $\boldsymbol{z}^a \leftarrow \boldsymbol{z}_i$.
  $\mathcal{Z}^+ \leftarrow \emptyset$.
  $\mathcal{Z}^- \leftarrow \emptyset$.
  for $(\boldsymbol{z}_j, y_j) \in \mathcal{D}_{\text{contrastive}} \setminus (\boldsymbol{z}_i, y_i)$ **do**
    if $y_i = y_j$ **then**
      $\mathcal{Z}^+ \leftarrow \mathcal{Z}^+ \bigcup \{\boldsymbol{z}_j\}$.
    else
      $\mathcal{Z}^- \leftarrow \mathcal{Z}^- \bigcup \{\boldsymbol{z}_j\}$.
    end if
  end for
  $\mathcal{L}_{\text{contrastive}} \leftarrow \mathcal{L}_{\text{contrastive}} + \ell_\tau^{\text{SCL}}(\boldsymbol{z}^a, \mathcal{Z}^+, \mathcal{Z}^-)$.
end for
$\mathcal{L}_{\text{contrastive}} \leftarrow \mathcal{L}_{\text{contrastive}}/B$.

3.2 Oversampling with PdFairDisCo

PdFairDisCo imposes a fairness constraint with distance covariance as an additional penalty on the objective function, in the same manner as FairDisCo. Although this softly constrained approach does not overly restrict the model architecture and the training algorithm, the resulting representations may still contain residual biases, i.e., remaining information about a sensitive attribute, which can be problematic in certain applications. In particular, we focus on downstream tasks where the values of a sensitive attribute are imbalanced in the training data: the samples corresponding to certain sensitive attribute values are fewer than others. Learning from such sensitive attribute-imbalanced data tends to excessively focus on the samples corresponding to majority sensitive attribute values and thus potentially amplifies biases even if the representations are nearly fair.

To address this challenging situation, we employ oversampling, a simple but widely applicable approach to the class imbalance problem [20]. Oversampling increases the

Algorithm 3 Computing a triplet loss for mini-batch data

Require: A mini-batch $\mathcal{D}_{\text{contrastive}} = \{(z_i, y_i)\}_{i=1}^{B}$, hyperparameters p and m.
Ensure: A triplet loss $\mathcal{L}_{\text{contrastive}}$.
$\mathcal{L}_{\text{contrastive}} \leftarrow 0$.
for $(z_i, y_i) \in \mathcal{D}_{\text{contrastive}}$ **do**
  $z^a \leftarrow z_i$.
  $\mathcal{Z}^+ \leftarrow \emptyset$.
  $\mathcal{Z}^- \leftarrow \emptyset$.
  for $(z_j, y_j) \in \mathcal{D}_{\text{contrastive}} \setminus (z_i, y_i)$ **do**
    if $y_i = y_j$ **then**
      $\mathcal{Z}^+ \leftarrow \mathcal{Z}^+ \bigcup \{z_j\}$.
    else
      $\mathcal{Z}^- \leftarrow \mathcal{Z}^- \bigcup \{z_j\}$.
    end if
  end for
  Select z^+ from $\mathcal{Z}^+$, and select z^- from $\mathcal{Z}^-$ uniformly at random.
  $\mathcal{L}_{\text{contrastive}} \leftarrow \mathcal{L}_{\text{contrastive}} + \ell_{p,m}^{\text{triplet}}(z^a, z^+, z^-)$.
end for
$\mathcal{L}_{\text{contrastive}} \leftarrow \mathcal{L}_{\text{contrastive}}/B$.

number of samples before training a prediction model by (1) replicating a randomly selected sample with a minority class and (2) adding the replicated sample to training data. This method balances classes and prevents the training process from being dominated by majority classes.

For our attribute imbalance problem in downstream tasks, there are two ways to apply oversampling to PdFairDisCo. In the first method, paired data $(\boldsymbol{x}, s)$ are transformed into a representation $\boldsymbol{z}$ with the encoder of PdFairDisCo, and then oversampling is performed for $\boldsymbol{z}$ if s takes a minority sensitive attribute value. In the second method, oversampling is first performed for $(\boldsymbol{x}, s)$ of a minority sensitive attribute value, and then the encoder transforms $(\boldsymbol{x}, s)$ into $\boldsymbol{z}$ as well as the replicated sample of $(\boldsymbol{x}, s)$.

Figure 2 shows the two oversampling methods for a binary sensitive attribute. Each circle or pentagon represents a data point: circles denote $(\boldsymbol{x}, s)$ and pentagons denote $\boldsymbol{z}$. Dark blue and orange indicate samples with majority and minority sensitive attribute values, respectively. A pair of closely overlapped points denotes $(\boldsymbol{x}, s)$ (or $\boldsymbol{z}$) and its replicated sample that takes the same value as the original. While the first method (upper row) performs oversampling after transforming with the encoder, the second method (lower row) does it beforehand with the transformation. Hence, we call the first method *post-oversampling* and the second method *pre-oversampling*. Compared to post-oversampling, pre-oversampling avoids adding exactly the same representations to the training data because $(\boldsymbol{x}, s)$ and its replicated sample are transformed into different representations due to independent sampling process.

To see the difference between the two methods more precisely, we consider obtaining two representations $\boldsymbol{z}$ and $\boldsymbol{z}^*$ from $(\boldsymbol{x}, s)$. In post-oversampling, $(\boldsymbol{x}, s)$ is first transformed into $\boldsymbol{z}$ by sampling from $q_\phi(\boldsymbol{z}|\boldsymbol{x}, s)$. This sampling can also be represented through the reparameterization trick:

$$\boldsymbol{z} = \boldsymbol{\mu}(\boldsymbol{x}, s) + \boldsymbol{\sigma}(\boldsymbol{x}, s) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{10}$$

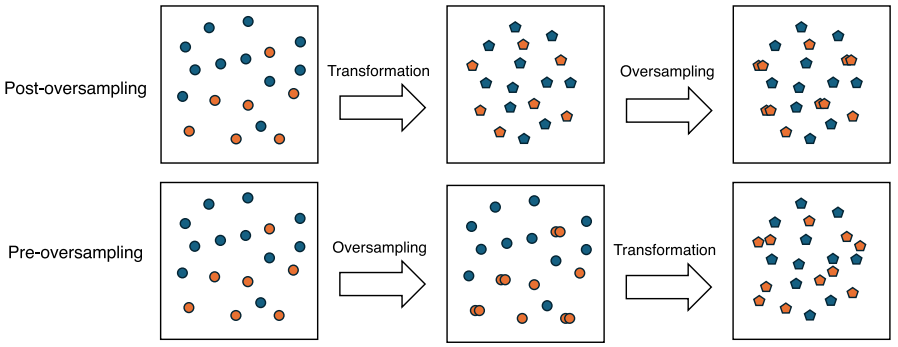


Fig. 2 Illustration of two oversampling methods using PdFairDisCo

where $\odot$ denotes the elementwise product, and $\boldsymbol{\epsilon} \in \mathbb{R}^{d_z}$ is Gaussian noise. Although the reparameterization trick is primarily required during training, we use it here for clarity: Eq. (10) indicates that the transformation can be viewed as adding Gaussian noise $\boldsymbol{\epsilon}$ scaled elementwise by $\boldsymbol{\sigma}(\boldsymbol{x}, s)$ to the mean vector $\boldsymbol{\mu}(\boldsymbol{x}, s)$. After sampling $\boldsymbol{z}$, the oversampling simply replicates it to obtain $\boldsymbol{z}^*$. As a result, the two representations are identical, i.e., $\boldsymbol{z} = \boldsymbol{z}^*$.

In contrast, pre-oversampling first replicates $(\boldsymbol{x}, s)$ to obtain $(\boldsymbol{x}^*, s^*)$. Then $(\boldsymbol{x}, s)$ is transformed into $\boldsymbol{z}$ by Eq. (10), while $(\boldsymbol{x}^*, s^*)$ is transformed into $\boldsymbol{z}^*$ using another independent sampling process:

$$\boldsymbol{z}^* = \boldsymbol{\mu}(\boldsymbol{x}^*, s^*) + \boldsymbol{\sigma}(\boldsymbol{x}^*, s^*) \odot \boldsymbol{\epsilon}^*, \quad \boldsymbol{\epsilon}^* \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{11}$$

where $\boldsymbol{\epsilon}^*$ is also Gaussian noise. Because $\boldsymbol{\epsilon}^*$ differs from $\boldsymbol{\epsilon}$, the resulting representations are no longer identical, i.e., $\boldsymbol{z} \neq \boldsymbol{z}^*$.

4 Experiments

We conducted experiments to verify the effectiveness of PdFairDisCo and the oversampling methods. In this section, we first explain the common experimental setup and present the experimental results.

4.1 Experimental Setup

We prepared four datasets for binary classification: the Adult dataset [21], the German dataset [22], the COMPAS dataset,[1] and the Default dataset [23]. Each dataset is widely used in fairness research [5, 7, 24] and consists of tabular data for multiple subjects. We used gender as the binary sensitive attribute (male or female) in all the experiments. The statistics of the datasets are shown in Table 1.

[1] https://github.com/propublica/compas-analysis.

Table 1 Statistics of the four datasets

	Number of samples	Number of features	Ratio of male to female
Adult	48,842	14	67:33
German	1000	20	69:31
COMPAS	7214	5	81:19
Default	30,000	23	40:60

Table 2 Confusion matrix

	The model predicts 1 ($\hat{y} = 1$)	The model predicts 0 ($\hat{y} = 0$)
True value is 1 ($y = 1$)	True positive (TP)	False negative (FN)
True value is 0 ($y = 0$)	False positive (FP)	True negative (TN)

To evaluate a representation learning model (e.g., PdFairDisCo), we split each dataset into three parts for training, validation, and a downstream task. Initially, we trained the model with the training data and selected the number of training epochs with the validation data. The last part for the downstream task was further divided into three splits. For the splits, we trained, validated and tested two classifiers whose set of input features $\boldsymbol{x}$ was transformed to a representation $\boldsymbol{z}$ beforehand by the encoder of the pre-trained representation learning model. The first was the label classifier to predict a class label y, and the other was the sensitive attribute classifier to predict a sensitive attribute s. We used random forests for both classifiers, following the experimental setup in [5]. The detailed procedure is described in Sect. A.1.

4.2 Evaluation Metrics

We evaluated the learned representations using four metrics: label accuracy, sensitive attribute accuracy, equal opportunity [25, Section 2.1], and equalized odds [25, Section 2]. Label accuracy and sensitive attribute accuracy are defined as the proportions of correct predictions made by classifiers trained to predict the label and sensitive attribute, respectively. High label accuracy indicates that the learned representations perform well in the downstream tasks, while low sensitive attribute accuracy (closer to 0.5 in binary classification) indicates that the representations contain little information about the sensitive attribute, and therefore achieve higher fairness.

To define equal opportunity and equalized odds, we first describe the confusion matrix, shown in Table 2, which defines the components used to compute true positive and false positive rates for each sensitive attribute.

Based on this, the true positive rate (TPR) and false positive rate (FPR) are defined as:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

Let $\mathrm{TPR}_{s=k}$ and $\mathrm{FPR}_{s=k}$ be the rates for the sensitive attribute $s = k$. Then the fairness metrics are calculated as:

$$\text{Equal opportunity} = |\mathrm{TPR}_{s=1} - \mathrm{TPR}_{s=0}|,$$
$$\text{Equalized odds} = |\mathrm{TPR}_{s=1} - \mathrm{TPR}_{s=0}| + |\mathrm{FPR}_{s=1} - \mathrm{FPR}_{s=0}|.$$

Smaller values indicate better fairness, with zero meaning identical classification behavior across sensitive attributes. We repeated independent training and evaluation 10 times and report the mean and standard deviation of the four metrics.

4.3 Evaluation of PdFairDisCo

4.3.1 Comparison with FairDisCo and VAEs

In this experiment, we compare PdFairDisCo with FairDisCo and VAEs to evaluate how our proposed approach can improve the existing one. The objective functions of FairDisCo and PdFairDisCo are given by Eq. (5) and (9) respectively. We set $\beta = 10^5$ for both by following the experiments in [5]. As mentioned above, we used the triplet loss for PdFairDisCo. To determine the strength of the triplet loss in PdFairDisCo, we selected α from $\{1, 2, \ldots, 9\}$ that achieved the lowest equalized odds on validation data in each downstream task. As a result, we chose $\alpha = 3$ for the Adult dataset, $\alpha = 1$ for the German dataset, $\alpha = 5$ for the COMPAS dataset, and $\alpha = 9$ for the Default dataset. We also evaluate a standard VAE (i.e. $\beta = 0$ and $\alpha = 0$) with the same architecture and experimental settings as PdFairDisCo for reference. Details of the training, validation and test procedure are provided in Sect. A.1.1.

Table 3 shows the mean and standard deviation of the four metrics for FairDisCo, PdFairDisCo, and the VAE. We first compare the VAE with the other two methods. We confirm that this simple VAE tends to yield worse fairness performance. In particular, the results of the VAE show high accuracy for sensitive attribute (e.g. 0.931 ± 0.002 for the Adult dataset). This means that we can easily reconstruct sensitive attributes from representations of the VAE. The results indicate that representation learning without fairness considerations could lead to bias in downstream tasks. Regarding label accuracy, the VAE generally outperforms FairDisCo but does not reach the performance of our PdFairDisCo. Because the VAE is less constrained, it may learn representations better suited for label classification than FairDisCo. However, the VAE does not match the predictive representations with PdFairDisCo. It should be noted that the VAE performs relatively well for the German dataset. The number of samples in the German dataset is much smaller than the other datasets as shown in Table 1. This could make the training process of more constrained FairDisCo and PdFairDisCo less stable.

Next, we compare PdFairDisCo with FairDisCo in more detail. We observe that PdFairDisCo consistently achieves higher label accuracy than FairDisCo across all datasets. In particular, PdFairDisCo outperforms FairDisCo for the Adult and COMPAS datasets by a large margin. These results support the effectiveness of PdFairDisCo, which enhances the representations and thus provides improved predictive performance in downstream tasks. Three fairness metrics exhibit different trends for each

Table 3 Comparison of four metrics among FairDisCo, PdFairDisCo, and a VAE

	FairDisCo	PdFairDisCo	VAE
Adult			
Label accuracy	0.798 ± 0.003	$\mathbf{0.835 \pm 0.003}$	0.813 ± 0.003
Sensitive attribute accuracy	0.646 ± 0.003	$\mathbf{0.640 \pm 0.003}$	0.931 ± 0.002
Equalized odds	0.329 ± 0.026	$\mathbf{0.066 \pm 0.023}$	0.099 ± 0.014
Equal opportunity	0.030 ± 0.007	$\mathbf{0.029 \pm 0.004}$	0.056 ± 0.006
German			
Label accuracy	0.687 ± 0.025	0.707 ± 0.019	$\mathbf{0.762 \pm 0.020}$
Sensitive attribute accuracy	$\mathbf{0.502 \pm 0.027}$	0.620 ± 0.025	0.662 ± 0.017
Equalized odds	0.439 ± 0.123	0.522 ± 0.110	$\mathbf{0.378 \pm 0.219}$
Equal opportunity	0.414 ± 0.119	0.481 ± 0.105	$\mathbf{0.333 \pm 0.211}$
COMPAS			
Label accuracy	0.585 ± 0.021	$\mathbf{0.632 \pm 0.015}$	0.629 ± 0.010
Sensitive attribute accuracy	0.781 ± 0.002	$\mathbf{0.778 \pm 0.002}$	0.999 ± 0.001
Equalized odds	0.170 ± 0.062	$\mathbf{0.154 \pm 0.049}$	0.399 ± 0.057
Equal opportunity	0.128 ± 0.057	$\mathbf{0.040 \pm 0.023}$	0.282 ± 0.061
Default			
Label accuracy	0.789 ± 0.004	$\mathbf{0.800 \pm 0.004}$	0.794 ± 0.003
Sensitive attribute accuracy	$\mathbf{0.557 \pm 0.009}$	0.561 ± 0.008	0.849 ± 0.004
Equalized odds	$\mathbf{0.026 \pm 0.019}$	0.027 ± 0.013	0.035 ± 0.013
Equal opportunity	0.019 ± 0.017	$\mathbf{0.018 \pm 0.011}$	0.027 ± 0.010

The best results are highlighted in bold

dataset. PdFairDisCo improves all fairness metrics in the Adult and COMPAS datasets, whereas the metrics worsen for the German dataset. The results are mixed for the Default dataset, though all the differences are quite small. There may be multiple causes that could explain these results. On the negative side, one might expect that PdFairDisCo inevitably suffers from a trade-off between label accuracy and fairness. Adding the triplet loss could result in underestimates of the distance covariance loss. On the positive side, improving label accuracy may also contribute to enhancing equal opportunity and equalized odds. The two metrics depend on the predictive performance of the label classifier, and more accurate label predictions with PdFairDisCo could reduce the disparities between the differences in TPRs and FPRs for different sensitive attribute values. Again, for the German dataset, the small sample size could make the training of more constrained PdFairDisCo unstable and the learned representations more susceptible to bias.

We note that the above experiments are not intended to achieve state-of-the-art performance, but are not conducted under overly restrictive settings. To support this, we also evaluated the label accuracy for a random forest trained directly on the input features. The results are 0.831 ± 0.001 for the Adult dataset, 0.838 ± 0.010 for the German dataset, 0.652 ± 0.007 for the COMPAS dataset, and 0.790 ± 0.002 for the Default dataset. By comparing to the results in Table 3, PdFairDisCo generally obtains

performance similar to the random forest (e.g. 0.835 ± 0.003 vs. 0.831 ± 0.001 for the Adult dataset), though we only observe a performance gap in the German dataset (i.e., 0.707 ± 0.019 vs. 0.838 ± 0.010).

To further verify the effectiveness of our approach, we compared PdFairDisCo with an MLP as an end-to-end baseline and with a variant that adds a cross-entropy (CE) loss term to FairDisCo instead of the contrastive loss. Details are provided in Sects. A.2.1 and A.2.2.

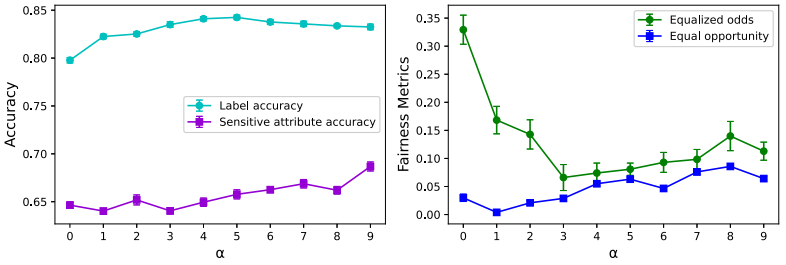
4.3.2 Exploring Roles of Triplet Loss

We further analyze the impact of the triplet loss in PdFairDisCo. Although we selected α based on equalized odds for validation data in Sect. 4.3.1, we also evaluated the other candidate values of α and investigated how the strength of the triplet loss influences both label accuracy and fairness in more depth.

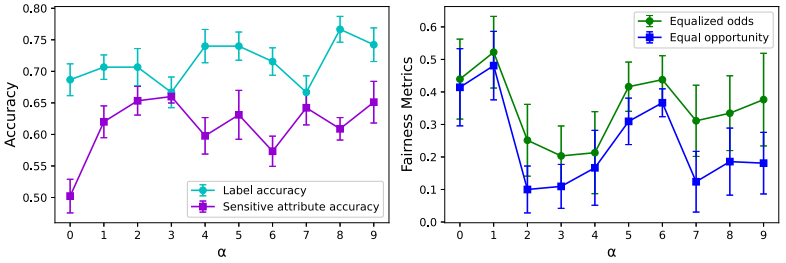
The results are shown in Fig. 3. The left side illustrates the results of label accuracy and sensitive attribute accuracy, and the right side depicts the results of equal opportunity and equalized odds. Note that the results of $\alpha = 0$ indicate the performance of FairDisCo. First, the results of label accuracy for nonzero α are generally better than $\alpha = 0$. We observe clear improvements in the Adult and COMPAS datasets and slight gains in the Default dataset for all comparisons. The German dataset also shows improvements except for certain two values of α. For the Adult and COMPAS datasets, the improvements decreased as α increased. It is consistent with our expectations; too large α could lead to an underestimate of the reconstruction loss and the regularization term in Eq. (9).

Next, we assess fairness metrics. We confirm that there are two trends for sensitive attribute accuracy. While the results worsen as α increases for the Adult and German datasets, the results show little change for the COMPAS and Default datasets. The trends of equal opportunity and equalized odds are more complicated. In the Adult and COMPAS datasets, the equal opportunity and equalized odds values tend to decrease initially and then increase as α increases, while those values tend to fluctuate with different values of α for the German and Default datasets. These diverse trends may be due to multiple causes with the triplet loss, as discussed in Sect. 4.3.1. In particular, equalized odds and equal opportunity could be affected by the positive and negative aspects. One possible implication of the results above is that a good model can be selected by choosing α based on either of fairness metrics. Since the results of label accuracy shows relatively stable improvements, choosing α according to fairness metrics would likely facilitate consistent improvement in all metrics, as demonstrated in Sect. 4.3.1.

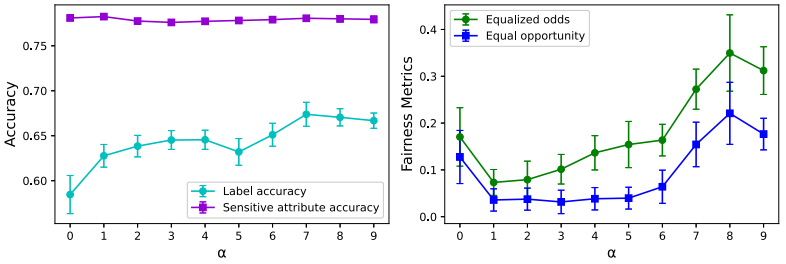
To further understand the learned representations, we visualize them using the t-SNE algorithm [26]. Figure 4 shows the representations of $\alpha = 0$ and $\alpha = 5$ in the Adult dataset. We can observe that the representations of the minority class (red plots) for $\alpha = 5$ tend to become more concentrated on the left-hand side than $\alpha = 0$. Note that we do not expect the representations to be clearly separated because they are subject to the prior distribution $p(z)$. Although t-SNE cannot fully preserve the original representations, the results modestly suggest that PdFairDisCo successfully learned more predictive representations.



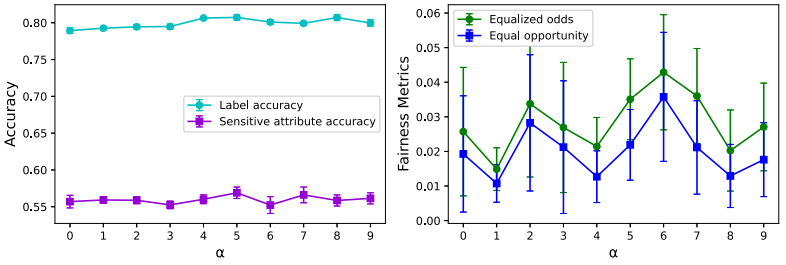
(a) Adult



(b) German

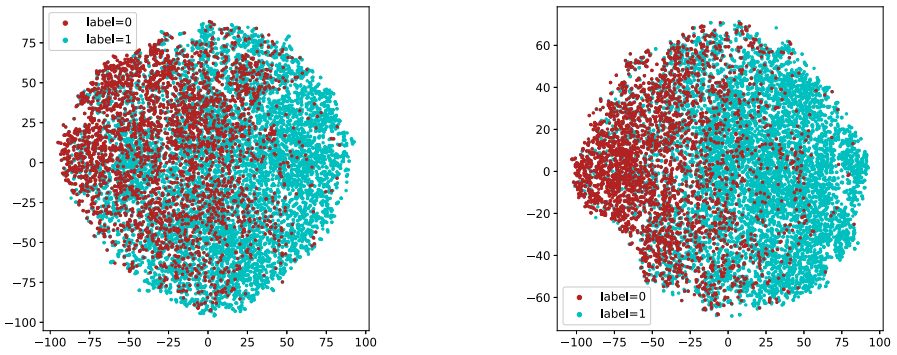


(c) COMPAS



(d) Default

Fig. 3 Results of FairDisCo ($\alpha = 0$) and PdFairDisCo on four metrics for different values of α



(a) FairDisCo (b) PdFairDisCo ($\alpha = 5$)

Fig. 4 Visualization of representations with t-SNE for the Adult dataset

4.3.3 Effect of Different Batch Sizes

Training with a triplet loss is known to be sensitive to batch size, as informative triplets—i.e., those for which the loss is non-zero and gradients are meaningful—must be sampled from within each mini-batch [19]. When such triplets are rare, the loss tends to be zero, and the representation learning may stagnate. To evaluate this effect, we compared the performance across batch sizes of 32, 64, 128, and 256. Other settings are same as the experiment in Sect. 4.3.1.

Figure 5 presents the results. While some variation was observed depending on the dataset, overall performance remained relatively stable across batch sizes. In the Adult and Default datasets, label accuracy showed little sensitivity to the batch size. The German and COMPAS datasets exhibited moderate fluctuations, but without a consistent advantage for larger or smaller batches. This robustness may be attributed to the number of classes in the datasets and the regularization effect of the Gaussian prior.

4.3.4 Comparison with SCL

In the previous experiments, we used the triplet loss as our contrastive loss. To examine whether the choice of contrastive losses affects the performance of PdFairDisCo, we additionally compare the triplet loss with the SCL. Specifically, we trained and evaluated an additional version of PdFairDisCo by replacing Eq. (8) with Eq. (7) and keeping all other experimental settings identical to those in Sect. 4.3.1.

Table 4 shows the experimental results. We can observe that the label accuracy and fairness metrics of the triplet loss were similar to or slightly better than those of the SCL in many cases, although the SCL showed performance gains in the German dataset. Since the SCL uses all positive and negative pairs within each batch, it could be more effective for the German dataset, whose number of samples is much smaller than the other datasets. Overall, these results suggest that our method is not overly sensitive to the specific choice of contrastive losses, at least in our experimental setup.

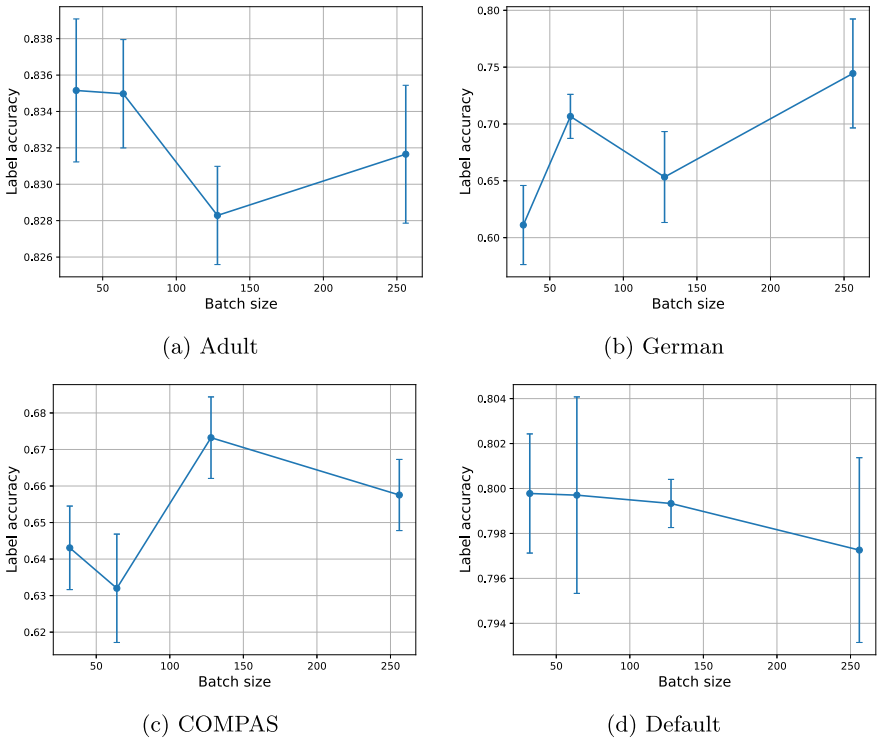


Fig. 5 Results of comparison among batch sizes 32, 64, 128, and 256

4.3.5 Transferability to Related Tasks

Standard representation learning seeks reusable representations across tasks. Although our PdFairDisCo learns representations that are specific to a target task, the learned representations can also be transferred to other related tasks. In the experiment below, we demonstrate that PdFairDisCo can be effective for transfer to one such related task.

We used the Adult dataset and considered a setting where the label y of age is not available at representation learning. Instead, we used another related label y' of income for representation learning with PdFairDisCo. We followed the same data splitting and evaluation protocol as in Sect. 4.3.1. In representation learning, we removed the feature corresponding to income from $\boldsymbol{x}$ and used it as the label y' to compute the contrastive loss of PdFairDisCo. We then trained random forests on the learned representations $\boldsymbol{z}$ obtained with FairDisCo and PdFairDisCo to predict y, and compared these results with a baseline in which a random forest was trained directly on $\boldsymbol{x}$ with the income feature removed.

As shown in Table 5, PdFairDisCo attains better label accuracy, equalized odds, and equal opportunity than FairDisCo. Compared to direct training of the random forest, the label accuracy is comparable, with a small improvement in the mean, while equalized odds and equal opportunity for both FairDisCo and PdFairDisCo are lower

Table 4 Comparison of four metrics between triplet loss and SCL

	Triplet loss	SCL
Adult		
Label accuracy	$\mathbf{0.835 \pm 0.003}$	0.832 ± 0.003
Sensitive attribute accuracy	$\mathbf{0.640 \pm 0.003}$	0.658 ± 0.004
Equalized odds	$\mathbf{0.066 \pm 0.023}$	0.102 ± 0.021
Equal opportunity	$\mathbf{0.029 \pm 0.004}$	0.031 ± 0.005
German		
Label accuracy	0.707 ± 0.019	$\mathbf{0.769 \pm 0.027}$
Sensitive attribute accuracy	$\mathbf{0.620 \pm 0.025}$	0.658 ± 0.027
Equalized odds	0.522 ± 0.110	$\mathbf{0.432 \pm 0.070}$
Equal opportunity	0.481 ± 0.105	$\mathbf{0.381 \pm 0.064}$
COMPAS		
Label accuracy	$\mathbf{0.632 \pm 0.015}$	0.625 ± 0.016
Sensitive attribute accuracy	$\mathbf{0.778 \pm 0.002}$	0.780 ± 0.007
Equalized odds	$\mathbf{0.154 \pm 0.049}$	0.157 ± 0.060
Equal opportunity	$\mathbf{0.040 \pm 0.023}$	0.113 ± 0.054
Default		
Label accuracy	$\mathbf{0.800 \pm 0.004}$	0.786 ± 0.004
Sensitive attribute accuracy	$\mathbf{0.561 \pm 0.008}$	0.575 ± 0.007
Equalized odds	$\mathbf{0.027 \pm 0.013}$	0.027 ± 0.016
Equal opportunity	$\mathbf{0.018 \pm 0.011}$	0.023 ± 0.014

The best results are highlighted in bold

Table 5 Comparison of four metrics among FairDisCo, PdFairDisCo, and random forest for transfer learning of the Adult dataset

	FairDisCo	PdFairDisCo	Random forest
Label accuracy	0.810 ± 0.002	$\mathbf{0.824 \pm 0.003}$	0.819 ± 0.002
Sensitive attribute accuracy	$\mathbf{0.645 \pm 0.004}$	0.662 ± 0.003	-
Equalized odds	0.140 ± 0.020	$\mathbf{0.060 \pm 0.018}$	0.174 ± 0.013
Equal opportunity	0.033 ± 0.005	$\mathbf{0.026 \pm 0.005}$	0.046 ± 0.003

The best results are highlighted in bold

than those of the random forest. It should be noted that sensitive attribute accuracy is omitted for the random forest, since it does not perform representation learning and thus has no representation z to use for predicting sensitive attribute s. Overall, these results suggest that the representations learned by PdFairDisCo can be reused for transfer learning across related labels in this setting.

4.4 Evaluation of Oversampling

In this experiment, we evaluate the effectiveness of the two oversampling methods proposed in Sect. 3.2. We performed similar experimental procedures as in Sect. 4.3 for PdFairDisCo with the two oversampling methods. α was fixed to the same value in Sect. 4.3.1, and we used the same training settings, e.g., optimizer and batch size, as in Sect. 4.3. Details of applying the oversampling methods are provided in Sect. A.1.2.

Table 6 shows the results for PdFairDisCo with the two oversampling methods. As a reference, we also copied from Table 3 the results of PdFairDisCo without oversampling, denoted as no-oversampling. We first compare the oversampling methods to no-oversampling. To begin with, we observe that oversampling has little effect on label accuracy. The differences between the oversampling methods and no-oversampling remain within 0.01 except for pre-oversampling in the German dataset. As we have repeatedly stated, the number of samples of the German dataset is much smaller than the others. Such a limited sample size could increase the instability of training algorithms and their predictions, and thus cause the drop in the label accuracy of pre-oversampling. Regarding fairness metrics, we find that the oversampling methods improve the performance over no-oversampling in multiple cases. Post-oversampling outperforms no-oversampling in 8 cases among 12 comparisons, while pre-oversampling consistently achieves the best sensitive attribute accuracy. These findings suggest that our proposed oversampling methods can generally contribute to improving fairness.

Next, we examine the differences between post-oversampling and pre-oversampling in more detail. Post-oversampling substantially improves the fairness metrics of no-oversampling. Pre-oversampling sometimes boosts the performance of fairness. In particular, pre-oversampling shows the best results across all datasets for sensitive attribute accuracy. However, there are also cases where pre-oversampling performs worse than the others. It yields the worst values for equalized odds and equal opportunity on three datasets and causes a substantial drop in label accuracy mentioned above. One possible reason for this instability is that pre-oversampling does not merely replicate data. As explained in Sect. 3.2, pre-oversampling produces two similar representations with independent noises drawn from the Gaussian distribution. While this data augmentation-like process can sometimes work well, the variability introduced by independent sampling may negatively affect the results in certain cases. In summary, if stable improvement in fairness metrics is desired, post-oversampling is recommended. If larger improvements of sensitive attribute accuracy are sought and some instability can be tolerated, pre-oversampling may also be considered.

5 Related Works

Recent studies on fair representation learning have explored various types of approach [4, 17, 24, 27, 28]. One common approach is to employ adversarial learning, in which two models are jointly trained so that one model tries to predict a sensitive attribute from a representation, and the other makes the prediction difficult [29, 30]. For example, FSNS [6] provides an improved approach that learns a discriminative

Table 6 Comparison of four metrics among the oversampling methods

	No-oversampling	Post-oversampling	Pre-oversampling
Adult			
Label accuracy	**0.835 ± 0.003**	0.832 ± 0.003	0.830 ± 0.003
Sensitive attribute accuracy	0.640 ± 0.003	0.633 ± 0.004	**0.603 ± 0.004**
Equalized odds	**0.066 ± 0.023**	0.082 ± 0.023	0.068 ± 0.021
Equal opportunity	**0.029 ± 0.004**	**0.029 ± 0.004**	0.043 ± 0.006
German			
Label accuracy	**0.707 ± 0.019**	0.696 ± 0.014	0.544 ± 0.038
Sensitive attribute accuracy	0.620 ± 0.025	0.560 ± 0.029	**0.509 ± 0.045**
Equalized odds	0.522 ± 0.110	0.506 ± 0.182	**0.410 ± 0.181**
Equal opportunity	0.481 ± 0.105	0.476 ± 0.164	**0.248 ± 0.127**
COMPAS			
Label accuracy	0.632 ± 0.015	0.634 ± 0.013	**0.638 ± 0.009**
Sensitive attribute accuracy	0.778 ± 0.002	0.765 ± 0.004	**0.517 ± 0.018**
Equalized odds	0.154 ± 0.049	**0.126 ± 0.045**	0.249 ± 0.053
Equal opportunity	0.040 ± 0.023	**0.025 ± 0.016**	0.106 ± 0.038
Default			
Label accuracy	0.800 ± 0.004	**0.802 ± 0.003**	0.790 ± 0.005
Sensitive attribute accuracy	0.561 ± 0.008	0.545 ± 0.006	**0.502 ± 0.010**
Equalized odds	**0.027 ± 0.013**	0.043 ± 0.014	0.067 ± 0.027
Equal opportunity	**0.018 ± 0.011**	0.030 ± 0.013	0.058 ± 0.028

The best results are highlighted in bold

representation for a class label while reducing information about sensitive attributes. While adversarial learning has also been used in many other contexts, it sometimes suffers from optimization instability and sensitivity to hyperparameters [15, 24]. In this paper, we explore FairDisCo and PdFairDisCo, VAE-based approaches that do not use adversarial learning.

Oversampling has been broadly used as a preprocessing step in the class imbalance problem. SMOTE [20] is a popular method that generates synthetic samples by interpolating between two data. SMOTE has been extended in various contexts, including fairness [7, 31]. In this study, we develop oversampling methods that simply replicate existing representations or produce similar representations with Gaussian noise rather than interpolation.

6 Conclusion

We proposed PdFairDisCo, an extension of FairDisCo for learning fair representations. The key idea of PdFairDisCo was to incorporate contrastive losses into the objective function in order to improve predictive performance in downstream tasks. We also introduced two oversampling methods for PdFairDisCo to address the imbalanced problem of sensitive attributes. Experimental results demonstrated that PdFairDisCo improved label accuracy and, in some cases, fairness metrics. The oversampling methods further improved fairness metrics without significantly degrading label accuracy.

Author Contributions Conceptualization: Tatsuya Yamada, Yoshinobu Kawahara; methodology: Tatsuya Yamada; formal analysis and investigation: Tatsuya Yamada, Takuya Konishi; writing—original draft preparation: Tatsuya Yamada; writing—review and editing: Takuya Konishi; resources: Yoshinobu Kawahara; Supervision: Yoshinobu Kawahara

Funding Open Access funding provided by The University of Osaka.

Data availability All datasets used in this study are publicly available. The sources are provided in the references and a GitHub repository, as cited in the main text.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Research involving human participants and/or animals Not applicable.

Informed consent Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix A Supplements to the Experiments

A.1 Experiment Procedure

In this section, we explain the concrete procedure of experiments mentioned in Sect. 4.

A.1.1 Evaluation of PdFairDisCo

In the experiment reported in Sect. 4.3, the entire dataset is first divided into three parts: $\mathcal{D}_{\text{train}}$ (70%), used for training PdFairDisCo; $\mathcal{D}_{\text{valid}}$ (15%), used for determining the number of training epochs of PdFairDisCo; and the remaining 15%, reserved for downstream tasks. The downstream portion is further split into: $\tilde{\mathcal{D}}_{\text{train}}$ (55%), $\tilde{\mathcal{D}}_{\text{valid}}$ (15%), and $\tilde{\mathcal{D}}_{\text{test}}$ (30%). Using these splits, the overall procedure is as follows:

1. Train PdFairDisCo on $\mathcal{D}_{\text{train}}$ using Algorithm 1 for each $\alpha \in \{1, \ldots, 9\}$ respectively. The number of training epochs is determined based on the validation loss computed on $\mathcal{D}_{\text{valid}}$.
2. For each trained encoder corresponding to $\alpha \in \{1, \ldots, 9\}$, we train a downstream label classifier on $\tilde{\mathcal{D}}_{\text{train}}$ and evaluate its fairness using equalized odds on $\tilde{\mathcal{D}}_{\text{valid}}$. The value of α that yields the lowest equalized odds is then selected.
3. We have the training dataset $\tilde{\mathcal{D}}' = \tilde{\mathcal{D}}_{\text{train}} \cup \tilde{\mathcal{D}}_{\text{valid}} = \{(\tilde{\boldsymbol{x}}_n, \tilde{s}_n, \tilde{y}_n)\}_{n=1}^{\tilde{N}'}$ for training and evaluation of downstream tasks. Note that $\tilde{N}' = \tilde{N}^{\text{train}} + \tilde{N}^{\text{valid}}$ where $\tilde{N}^{\text{train}}$ ($\tilde{N}^{\text{valid}}$) means the number of $\tilde{\mathcal{D}}_{\text{train}}$ ($\tilde{\mathcal{D}}_{\text{valid}}$).
4. Fixing α to the selected value from step 2, apply the encoder f_ϕ to $\{(\tilde{\boldsymbol{x}}_n, \tilde{s}_n)\}_{n=1}^{\tilde{N}'}$ to obtain the representations $\{\tilde{\boldsymbol{z}}_n\}_{n=1}^{\tilde{N}'}$. Then, we have the set of tuple $\{(\tilde{\boldsymbol{z}}_n, \tilde{s}_n, \tilde{y}_n)\}_{n=1}^{\tilde{N}'}$.
5. Train the label classifier with $\{(\tilde{\boldsymbol{z}}_n, \tilde{y}_n)\}_{n=1}^{\tilde{N}'}$. In the same way, train the sensitive attribute classifier with $\{(\tilde{\boldsymbol{z}}_n, \tilde{s}_n)\}_{n=1}^{\tilde{N}'}$.
6. Encode $\tilde{\mathcal{D}}_{\text{test}}$ with the encoder f_ϕ and evaluate the trained classifiers using both accuracy and fairness metrics.

Further implementation details, including the hidden layer sizes and the number of training epochs for both PdFairDisCo and downstream classifiers, are provided in Appendix A.3.

A.1.2 Evaluation of Oversampling

The three methods compared in Sect. 4.4 are no-oversampling, post-oversampling, and pre-oversampling. Both post-oversampling and pre-oversampling apply random oversampling to correct the imbalance in the number of data. In post-oversampling, oversampling is performed after transformation, whereas in pre-oversampling, it is applied beforehand while avoiding duplication of the oversampled data. The no-oversampling method is included for comparison and does not apply any oversampling.

No-oversampling corresponds to step 1 to 6 in Sect. A.1.1. Post-oversampling investigates the effect of the first method described in Sect. 3.2. In this method, after

executing step 4 in no-oversampling to obtain the representations, random oversampling is applied using the procedure in step A. This process is repeated until the number of tuples (z, s, y) with the minority and majority values of the sensitive attribute becomes equal.

A Let s_m be the minority value of the sensitive attribute.
Sampling a tuple $(\tilde{z}_j, \tilde{s}_j, \tilde{y}_j)$ uniformly at random, where $j \in \{1, 2, \ldots, \tilde{N}'\}$, from the set $\{(\tilde{z}_n, \tilde{s}_n = s_m, \tilde{y}_n)\}_{n=1}^{\tilde{N}'}$. Then, add the sampled tuple to the set and update it accordingly. Each time this step is executed, the number of training samples $\tilde{N}'$ is incremented by one.

Similarly, pre-oversampling examines the effect of the second method described in Sect. 3.2. In this method, before executing step 4 in no-oversampling, random oversampling is applied using the procedure in step B. This oversampling is repeated until the number of training samples for each value of the sensitive attribute becomes equal.

B Let s_m be the minority value of the sensitive attribute.
Sampling a tuple $(\tilde{\boldsymbol{x}}_j, \tilde{s}_j{=}s_m, \tilde{y}_j)$ uniformly at random, where $j \in \{1, 2, \ldots, \tilde{N}'\}$, from $\tilde{\mathcal{D}}'$. Then, add the sampled tuple to the set and update it accordingly. Each time this step is executed, the number of training samples $\tilde{N}'$ is incremented by one.

A.2 Additional Comparisons

A.2.1 Comparison with End-to-End Baseline

As discussed in Sect. 3.1, although end-to-end approaches are widely used in task-specific settings, additional data are often obtained after initial training in practice. In such situations, it can be preferable to update the model by leveraging only newly obtained data; in this setting, we treat this update as the downstream task.

We describe several features of our approach compared to retraining end-to-end models under the above setting. First, our framework allows a broader range of models for the downstream task to be considered. For example, our experiments use random forests, and tree-based classifiers have been reported to sometimes outperform deep models in certain settings [32]. Second, in terms of fairness, another feature of our approach is that fairness constraints do not need to be explicitly handled when training a downstream classifier on additional data. Since our PdFairDisCo transforms inputs into fair representations, we can use a broader class of classifiers without fairness-aware training. In contrast, when retraining an end-to-end model under fairness considerations, fairness must still be carefully considered. This can increase the practical effort required to retrain models under fairness considerations.

We additionally compare PdFairDisCo with an end-to-end baseline. An MLP with the same structure as the encoder of PdFairDisCo is used, and jointly optimized predictive performance and fairness using the following objective:

$$\mathcal{L}_{\text{end-to-end}} = \mathcal{L}_{\text{CE}} + \beta \mathcal{L}_{\text{fair}}\,, \tag{12}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the CE loss for predicting the label and $\mathcal{L}_{\mathrm{fair}}$ is a fairness penalty. We do not employ distance covariance for the end-to-end baseline. Distance covariance relies on a closed-form expression under the assumption that $q_\phi(\boldsymbol{z}|\boldsymbol{x}, s)$ follows a Gaussian distribution, and it is not straightforward to apply it to the end-to-end baseline without introducing additional distributional assumptions. Instead, we use maximum mean discrepancy (MMD) as $\mathcal{L}_{\mathrm{fair}}$. MMD is a differentiable, sample-based penalty that can be computed from mini-batch samples, and it has been used to promote fairness in prior work on VAE-based fair representation learning [17]. In addition, FairDisCo [5] discusses MMD as a related loss that, like distance covariance, aims to promote independence. In our implementation, we compute MMD on the intermediate representations $\{\boldsymbol{z}_i\}$ (the MLP features before the final linear layer) and define $\mathcal{L}_{\mathrm{fair}}$ as the average MMD between sensitive attribute values within each mini-batch, using a radial basis function kernel.

We followed the same data split and evaluation protocol as in Sect. A.1.1. Concretely, for the end-to-end baseline, we trained the model using the portions corresponding to $\mathcal{D}_{\mathrm{train}}$, $\tilde{\mathcal{D}}_{\mathrm{train}}$, and $\tilde{\mathcal{D}}_{\mathrm{valid}}$, selected hyperparameters based on $\mathcal{D}_{\mathrm{valid}}$, and reported the final performance on $\tilde{\mathcal{D}}_{\mathrm{test}}$. We tuned β by the same selection procedure as in Sect. 4.3.1 from $\beta \in \{0.1, 1, 5, 10, 30, 50, 100\}$. For the German dataset, we observed unstable behavior for larger β values, and we therefore excluded such unstable configurations and selected β from $\{0.1, 1, 5, 10\}$. As a result, we chose $\beta = 50$ for the Adult dataset, $\beta = 5$ for the German dataset, $\beta = 50$ for the COMPAS dataset, and $\beta = 1$ for the Default dataset.

The results are shown in Table 7. We observe different trends across dataset. Except for the German dataset, the differences are generally small, and neither method is consistently superior across all metrics. For the Adult and the Default datasets, the end-to-end baseline achieves slightly higher label accuracy and sensitive attribute accuracy, whereas PdFairDisCo attains lower equalized odds and equal opportunity. For the COMPAS dataset, PdFairDisCo performs slightly better on label accuracy, equalized odds, and equal opportunity, while the end-to-end baseline achieves better sensitive attribute accuracy. In contrast, on the German dataset, the end-to-end baseline performs better across all metrics, suggesting that the two-stage pipeline may be less effective when the sample size is small. In summary, PdFairDisCo achieves performance comparable to the end-to-end baseline on Adult, COMPAS, and Default, while a notable gap is observed on German. As noted in Sect. 4, the German dataset contains relatively few samples and tends to exhibit different behavior from the other datasets.

A.2.2 Comparison with CE Loss

In our task-specific setting, label information is available. Therefore, in addition to contrastive losses, one can alternatively use a standard classification loss such as CE loss. To examine how this direct objective compares with contrastive losses, we introduce an additional comparison model, *FairDisCo-CE*, and evaluate it under the same experimental protocol as in Sect. 4.3.1.

Table 7 Comparison of four metrics between PdFairDisCo and end-to-end baseline

	PdFairDisCo	End-to-end
Adult		
Label accuracy	0.835 ± 0.003	$\mathbf{0.845 \pm 0.003}$
Sensitive attribute accuracy	0.640 ± 0.003	$\mathbf{0.630 \pm 0.012}$
Equalized odds	$\mathbf{0.066 \pm 0.023}$	0.097 ± 0.046
Equal opportunity	$\mathbf{0.029 \pm 0.004}$	0.035 ± 0.021
German		
Label accuracy	0.707 ± 0.019	$\mathbf{0.815 \pm 0.018}$
Sensitive attribute accuracy	0.620 ± 0.025	$\mathbf{0.500 \pm 0.055}$
Equalized odds	0.522 ± 0.110	$\mathbf{0.254 \pm 0.102}$
Equal opportunity	0.481 ± 0.105	$\mathbf{0.152 \pm 0.070}$
COMPAS		
Label accuracy	$\mathbf{0.632 \pm 0.015}$	0.624 ± 0.032
Sensitive attribute accuracy	0.778 ± 0.002	$\mathbf{0.759 \pm 0.022}$
Equalized odds	$\mathbf{0.154 \pm 0.049}$	0.167 ± 0.062
Equal opportunity	$\mathbf{0.040 \pm 0.023}$	0.063 ± 0.053
Default		
Label accuracy	0.800 ± 0.004	$\mathbf{0.808 \pm 0.003}$
Sensitive attribute accuracy	0.561 ± 0.008	$\mathbf{0.545 \pm 0.013}$
Equalized odds	$\mathbf{0.027 \pm 0.013}$	0.033 ± 0.016
Equal opportunity	$\mathbf{0.018 \pm 0.011}$	0.023 ± 0.015

The best results are highlighted in bold

FairDisCo-CE augments the FairDisCo objective with a CE loss term computed from a linear prediction head attached to the representation z:

$$\mathcal{L}_{\text{FairDisCo-CE}} = \mathcal{L}_{\text{FairDisCo}} + \alpha \mathcal{L}_{\text{CE}}, \tag{13}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss for predicting the label from z, and α controls the strength of the CE term. This can be viewed as FairDisCo-CE replacing the contrastive loss term in PdFairDisCo with a CE term. We tuned $\alpha \in \{1, 2, 3, 4, 5\}$ using the same selection procedure as in Sect. 4.3.1. Consequently, $\alpha = 1$ was selected for the Adult, German, and Default datasets, and $\alpha = 5$ was chosen for the COMPAS dataset.

Table 8 summarizes the experimental results. First, PdFairDisCo achieves higher label accuracy than FairDisCo-CE across all datasets. Intuitively, contrastive losses encourage representations z of samples that share the same label to be closer than those with different labels, which can promote more discriminative representations and thereby improve predictive performance in our experiments. Next, the fairness metrics vary across datasets. PdFairDisCo performs better than FairDisCo-CE on the COMPAS dataset in both equalized odds and equal opportunity, whereas FairDisCo-CE yields better fairness results for the German and Default datasets. Overall, these results suggest that contrastive losses are consistently more effective for predictive

Table 8 Comparison of four metrics between PdFairDisCo and FairDisCo-CE

	PdFairDisCo	FairDisCo-CE
Adult		
Label accuracy	$\mathbf{0.835 \pm 0.003}$	0.815 ± 0.003
Sensitive attribute accuracy	$\mathbf{0.640 \pm 0.003}$	0.641 ± 0.004
Equalized odds	$\mathbf{0.066 \pm 0.023}$	0.194 ± 0.015
Equal opportunity	0.029 ± 0.004	$\mathbf{0.005 \pm 0.003}$
German		
Label accuracy	$\mathbf{0.707 \pm 0.019}$	0.689 ± 0.034
Sensitive attribute accuracy	0.620 ± 0.025	$\mathbf{0.607 \pm 0.037}$
Equalized odds	0.522 ± 0.110	$\mathbf{0.359 \pm 0.126}$
Equal opportunity	0.481 ± 0.105	$\mathbf{0.276 \pm 0.102}$
COMPAS		
Label accuracy	$\mathbf{0.632 \pm 0.015}$	0.617 ± 0.016
Sensitive attribute accuracy	$\mathbf{0.778 \pm 0.002}$	$\mathbf{0.778 \pm 0.003}$
Equalized odds	$\mathbf{0.154 \pm 0.049}$	0.290 ± 0.051
Equal opportunity	$\mathbf{0.040 \pm 0.023}$	0.176 ± 0.047
Default		
Label accuracy	$\mathbf{0.800 \pm 0.004}$	0.787 ± 0.003
Sensitive attribute accuracy	0.561 ± 0.008	$\mathbf{0.551 \pm 0.008}$
Equalized odds	0.027 ± 0.013	$\mathbf{0.018 \pm 0.010}$
Equal opportunity	0.018 ± 0.011	$\mathbf{0.008 \pm 0.008}$

The best results are highlighted in bold

performance in our experiments, while their impact on fairness is dataset-dependent and does not show a clear advantage over using a CE term.

A.3 Experimental Settings

The hyperparameters of the model and training for FairDisCo and the downstream tasks were generally determined by referring to the experiments conducted by Liu et al. Only the number of training epochs was determined using the validation dataset. The triplet loss parameters are the default values of TripletMarginLoss[2] from Pytorch [33]. Also, the SCL parameters follow the default settings provided in the official implementation.[3] For the downstream tasks, we used RandomForestClassifier[4] from scikit-learn [34], with default values for all parameters. The specific values are shown in Table 9.

[2] https://pytorch.org/docs/stable/generated/torch.nn.TripletMarginLoss.html.

[3] https://github.com/HobbitLong/SupContrast.

[4] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

Table 9 Hyperparameters used in experiments

Description	Value
Number of layers in encoder and decoder	2
Dimension of hidden layers in encoder and decoder	64
Dimension of the representations	8
Batch size	64
Training epochs (German dataset)	400
Training epochs (other dataset)	1000
Optimizer	Adam
Training rate	10^{-3}
Margin of the triplet loss	1.0
Degree of norm of the triplet loss	2
Temperature scaling of the SCL	0.07

References

1. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Comput. Surv. **54**(6), 115–111535 (2022). https://doi.org/10.1145/3457607
2. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And It's Biased Against Blacks. ProPublica, New York (2016)
3. Lambrecht, A., Tucker, C.: Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of STEM career ads. Manag. Sci. **65**(7), 2966–2981 (2019). https://doi.org/10.1287/MNSC.2018.3093
4. Creager, E., Madras, D., Jacobsen, J., Weis, M.A., Swersky, K., Pitassi, T., Zemel, R.S.: Flexibly fair representation learning by disentanglement. In: Proceedings of the 36th International Conference on Machine Learning, pp. 1436–1445 (2019)
5. Liu, J., Li, Z., Yao, Y., Xu, F., Ma, X., Xu, M., Tong, H.: Fair representation learning: an alternative to mutual information. In: KDD '22: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 18661–18673 (2022). https://doi.org/10.1145/3534678.3539302
6. Jang, T., Gao, H., Shi, P., Wang, X.: Achieving fairness through separability: a unified framework for fair representation learning. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 28–36 (2024)
7. Chakraborty, J., Majumder, S., Menzies, T.: Bias in machine learning software: why? How? What to do? In: ESEC/FSE '21: 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 429–440 (2021). https://doi.org/10.1145/3468264.3468537
8. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations (2014)
9. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 539–546 (2005). https://doi.org/10.1109/CVPR.2005.202
10. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1735–1742 (2006). https://doi.org/10.1109/CVPR.2006.100
11. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1386–1393 (2014). https://doi.org/10.1109/CVPR.2014.180
12. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015). https://doi.org/10.1109/CVPR.2015.7298682

13. Vygon, R., Mikhaylovskiy, N.: Learning efficient representations for keyword spotting with triplet loss. In: Speech and Computer, pp. 773–785 (2021). https://doi.org/10.1007/978-3-030-87802-3_69
14. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: Advances in Neural Information Processing Systems 33, pp. 18661–18673 (2020)
15. Moyer, D., Gao, S., Brekelmans, R., Galstyan, A., Steeg, G.V.: Invariant representations without adversarial training. In: Advances in Neural Information Processing Systems 31, pp. 9102–9111 (2018)
16. Gálvez, B.R., Thobaben, R., Skoglund, M.: A variational approach to privacy and fairness. In: IEEE Information Theory Workshop, pp. 1–6 (2021). https://doi.org/10.1109/ITW48936.2021.9611429
17. Louizos, C., Swersky, K., Li, Y., Welling, M., Zemel, R.S.: The variational fair autoencoder. In: 4th International Conference on Learning Representations (2016)
18. Székely, G.J., Rizzo, M.L., Bakirov, N.K.: Measuring and testing dependence by correlation of distances. Ann. Stat. (2007). https://doi.org/10.1214/009053607000000505
19. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. Preprint at https://arxiv.org/abs/1703.07737 (2017)
20. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002). https://doi.org/10.1613/JAIR.953
21. Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository (1996). https://doi.org/10.24432/C5XW20
22. Hofmann, H.: Statlog (german credit data). UCI Machine Learning Repository (1994). https://doi.org/10.24432/C5NC77
23. Yeh, I.-C.: Default of credit card clients. UCI Machine Learning Repository (2009). https://doi.org/10.24432/C55S3H
24. Oh, C., Won, H., So, J., Kim, T., Kim, Y., Choi, H., Song, K.: Learning fair representation via distributional contrastive disentanglement. In: KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1295–1305 (2022). https://doi.org/10.1145/3534678.3539232
25. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: Advances in Neural Information Processing Systems 29, pp. 3315–3323 (2016)
26. Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(86), 2579–2605 (2008)
27. Song, J., Kalluri, P., Grover, A., Zhao, S., Ermon, S.: Learning controllable fair representations. In: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, pp. 2164–2173 (2019)
28. Sarhan, M.H., Navab, N., Eslami, A., Albarqouni, S.: Fairness by learning orthogonal disentangled representations. In: Computer Vision - ECCV 2020. Lecture Notes in Computer Science, pp. 746–761 (2020). https://doi.org/10.1007/978-3-030-58526-6_44
29. Madras, D., Creager, E., Pitassi, T., Zemel, R.S.: Learning adversarially fair and transferable representations. In: Proceedings of the 35th International Conference on Machine Learning, pp. 3381–3390 (2018)
30. Beutel, A., Chen, J., Zhao, Z., Chi, E.H.: Data decisions and theoretical implications when adversarially learning fair representations. In: 4th Workshop on Fairness, Accountability, and Transparency in Machine Learning (2017)
31. Fernández, A., García, S., Herrera, F., Chawla, N.V.: SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. J. Artif. Intell. Res. **61**, 863–905 (2018). https://doi.org/10.1613/JAIR.1.11192
32. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? In: Advances in Neural Information Processing Systems 35 (2022)
33. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E.Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035 (2019)
34. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**(85), 2825–2830 (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.