FLAC: Fairness-Aware Representation Learning by Suppressing Attribute-Class Associations

Ioannis Sarridis, Christos Koutlis, Symeon Papadopoulos, and Christos Diou, *Member, IEEE*

***Abstract*—Bias in computer vision systems can perpetuate or even amplify discrimination against certain populations. Considering that bias is often introduced by biased visual datasets, many recent research efforts focus on training fair models using such data. However, most of them heavily rely on the availability of protected attribute labels in the dataset, which limits their applicability, while label-unaware approaches, i.e., approaches operating without such labels, exhibit considerably lower performance. To overcome these limitations, this work introduces FLAC, a methodology that minimizes mutual information between the features extracted by the model and a protected attribute, without the use of attribute labels. To do that, FLAC proposes a sampling strategy that highlights underrepresented samples in the dataset, and casts the problem of learning fair representations as a probability matching problem that leverages representations extracted by a bias-capturing classifier. It is theoretically shown that FLAC can indeed lead to fair representations, that are independent of the protected attributes. FLAC surpasses the current state-of-the-art on Biased-MNIST, CelebA, and UTKFace, by 29.1%, 18.1%, and 21.9%, respectively. Additionally, FLAC exhibits 2.2% increased accuracy on ImageNet-A and up to 4.2% increased accuracy on Corrupted-Cifar10. Finally, in most experiments, FLAC even outperforms the bias label-aware state-of-the-art methods.**

***Index Terms*—Bias mitigation, fairness, mutual information.**

I. Introduction

DURING the past decade, AI models have demonstrated exceptional performance in a growing number of application areas; however, there have also been numerous incidents where failures in the AI systems have disproportionately affected certain individuals or groups of people [1]. Bias in Artificial Intelligence (AI) typically refers to AI systems that demonstrate discriminatory behavior (e.g., high errors) against groups or populations w.r.t. certain *protected* or *sensitive* attributes that in several jurisdictions are legally protected from discrimination, e.g., race, gender, age, and religion. This makes fairness and non-discrimination an important ethical and often legal requirement for AI models, and a prerequisite for their wide adoption and use in many real-world applications [2], [3], [4], [5].

Bias can emerge in several Computer Vision (CV) systems, often in applications involving the processing of face images but also in other domains where visual data is involved [6]. Many research efforts focusing on face verification fairness [7], [8], [9] highlight that AI models tend to be inaccurate for black people, especially for black women [10]. In addition, similar biased behavior against under-represented groups has been noticed in AI systems for face recognition [10], facial expression recognition [11], and facial attribute analysis [12].

Taking into account the importance of fairness in AI, several approaches for learning fair representations have been proposed in recent years. The underlying idea of such approaches is that if the representations of samples belonging to different groups w.r.t. a protected attribute are similar, then the classifier built on top of them will inevitably make decisions irrespective of the group to which the samples belong. This serves as a means for achieving demographic parity (statistical parity) between the groups of different protected attribute values [13]. Many approaches make use of the protected attribute labels (also known as bias labels in the literature) provided in the training set [14], [15], [16], [17]. We refer to those as Bias Label-Aware (BL-A) methods. Despite the effectiveness of such methods in terms of bias mitigation, they can only be applied in a narrow range of problems and datasets, due to their reliance on the availability of protected attribute labels. To overcome this limitation, several recent efforts employ the representations derived by a bias-capturing classifier for mitigating the bias without being aware of the protected attribute labels [18], [19], [20], [21]. We refer to these methods as Bias Label-Unaware (BL-U). The bias-capturing classifier is a model that attempts to encapsulate information for the protected attribute. In addition, BL-U approaches can be applied in cases where bias is not categorical. For instance, representations with texture bias can be derived by a CNN with small receptive fields [20]. However, the performance of BL-U approaches has been shown to lag considerably behind the one of BL-A methods [16], [17].

Mutual Information (MI) can be used to measure the dependence between the representation provided by the main model and the protected attribute values. By minimizing MI during representation learning, one can deter the main model from using

Received 28 April 2023; revised 5 September 2023; accepted 24 October 2024. Date of publication 28 October 2024; date of current version 9 January 2025. This work was supported in part by the EU Horizon Europe projects MAMMOth under Grant 101070285, in part by the ELIAS under Gant 101120237, and in part by the EU H2020 project MediaVerse under Grant 957252. Recommended for acceptance by T. Tommasi. *(Corresponding author: Ioannis Sarridis.)*

Ioannis Sarridis is with the Department of Informatics and Telematics, Harokopio University of Athens, 17676 Athens, Greece, and also with the Information Technologies Institute, Centre for Research and Technology Hellas, 57001 Thessaloniki, Greece (e-mail: gsarridis@iti.gr).

Christos Koutlis and Symeon Papadopoulos are with the Information Technologies Institute, Centre for Research and Technology Hellas, 57001 Thessaloniki, Greece (e-mail: ckoutlis@iti.gr; papadop@iti.gr).

Christos Diou is with the Department of Informatics and Telematics, Harokopio University of Athens, 17676 Athens, Greece (e-mail: cdiou@hua.gr).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3487254, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3487254

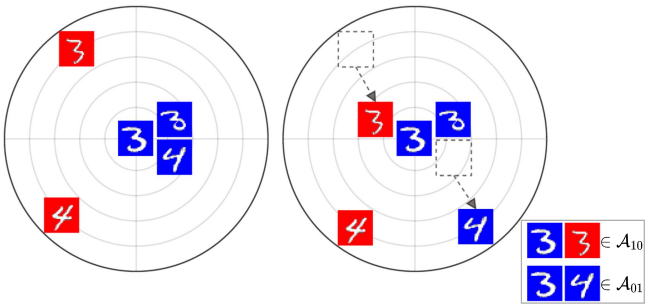


Fig. 1. Distances between the central sample and the rest of the samples belonging to Biased-MNIST, a dataset that demonstrates a strong association between the target labels (i.e., *digits*) and the protected attributes (i.e., *colors*). Subfigure on the left shows that using the standard task-specific loss introduces bias to the representations which FLAC, the proposed approach, can effectively mitigate as shown on the right. In particular, FLAC focuses on situations where task-specific losses are susceptible to data bias, namely pairs with the same target label but different protected attribute values (i.e., $\mathcal{A}_{10}$) and pairs with different target labels and the same protected attribute value (i.e., $\mathcal{A}_{01}$).

a given protected attribute to predict the label. In this way, fairness can be achieved through a group-invariant representation, which is known to guarantee accuracy parity between the groups defined by the protected attribute [13]. However, solving the above MI minimization problem would require access to the protected attributes (i.e., would need a BL-A method). In this paper we introduce FLAC, a BL-U approach that leverages the representations of a bias-capturing classifier to force an initial potentially biased model to learn fairer representations. In particular, we cast the reduction of the MI minimization problem into a simpler probability matching problem between the similarity distributions of the main model and the bias-capturing classifier. This turns out to be an effective means of disassociating the target representation from the bias-capturing model and, as a result, from the protected attributes. To this end, the proposed method leverages the pairs of samples for which a typical task-specific loss is prone to bias, namely samples sharing either only targets or only protected attributes (see Fig. 1). This is not the case for previous BL-U methods that ignore the importance of a proper selection process, thereby impeding the bias mitigation effectiveness. Furthermore, it is theoretically justified that FLAC can minimize the mutual information between the main model representation and protected attribute.

In summary, the main contributions of this paper are the following:

- A fairness-aware representation learning approach that enables a model to learn fair representations w.r.t. a protected attribute by leveraging the representations of a bias-capturing classifier, without using the protected attribute labels.
- A condition for selecting the under-represented sample pairs that can contribute most effectively to the bias mitigation procedure.
- A wide comparative analysis involving 9 state-of-the-art approaches, 5 benchmark datasets, namely Biased-MNIST [20], CelebA [22], UTK-Face [23], Corrupted-Cifar10 [24] and 9-Class ImageNet [20]. In all the conducted experiments, FLAC surpasses the current BL-U state-of-the-art by achieving +29.1%, +18.1%, +21.9%, +4.2%, and +2.2% accuracy on Biased-MNIST, CelebA, UTKFace, Corrupted-Cifar10, and 9-Class ImageNet, respectively. Moreover, in most experiments, FLAC even outperforms the BL-A state-of-the-art.

The code is available at https://github.com/gsarridis/FLAC.

II. Related Work

In recent years, many approaches have emerged for mitigating bias in neural networks. Some approaches focus on debiasing the training data by balancing the available data across the different groups of interest [25], using more data sources [26], applying data augmentation [27], and collecting more data [28], [29]. However, such data-oriented approaches do not provide the flexibility required in cases where fairness requirements are not static (e.g., the need for considering a new protected attribute), while data availability may also be limited. To address these shortcomings, many efforts focus on developing approaches to mitigate bias while training a model using biased data (i.e., in-processing approaches [30]). To achieve this, many works propose ensembling [18], [31] or adversarial frameworks [14], [32], [33], [34], [35], [36], contrastive learning-based approaches [16], [17], and regularization terms [15], [16], [19]. These in-processing bias mitigation approaches can be divided into two primary categories that significantly impact their range of applicability and performance: methods that necessitate access to the protected attribute labels, i.e., BL-A, and methods that do not require such labels, i.e., BL-U.

a) Bias mitigation with protected attribute labels: In [14] an adversarial approach is proposed, namely Learning Not to Learn (LNL), based on the MI between feature embeddings and protected attribute labels, that penalizes the model if it is capable of predicting the protected attributes. Domain-Independent (DI) [31] suggests multiple classifier heads, one per domain (i.e., bias type), for mitigating the bias in the feature space. The approach of Entangling and Disentangling deep representations (EnD) [15] proposes a regularization process that tries to entangle the feature vectors of samples with the same target class and disentangle the features representations of samples that share the same protected attribute label. Given the representations derived by the main model, FairKL [17] aims to match the distances of positive (negative) bias-conflicting and bias-aligned samples from the anchor, where bias-conflicting and bias-aligned denote the under-represented and the over-represented groups, respectively. When protected attribute labels are not available, the similarities between the biased features are utilized to assign weights to the FairKL regularizer. By contrast to FairKL, FLAC solves a probability matching problem between the representations of a bias-capturing classifier and the main model. Finally, the Bias-Contrastive and Bias-Balance (BC-BB) method [16] constitutes a contrastive learning-based approach for bias mitigation. Bias-Contrastive (BC) encourages the main model to pull the samples with the same target but different protected attributes closer in the feature space while ignoring the bias-aligned samples with different target labels. Bias-Balance (BB) can be used in combination with BC to further mitigate the bias by optimizing the model toward the data distribution. Although these methods demonstrate state-of-the-art performance in bias

mitigation, they require the protected attribute labels to work. This constitutes a crucial limitation, as they can not be applied in many real-world settings, where protected attribute labels are not available, or even worse, in cases where the bias is introduced by non-categorical visual attributes (e.g., texture).

b) Bias mitigation without protected attribute labels: The significant limitation described above, led the research community to explore methods that can address bias without being aware of the protected attribute labels. In [18], the Learned-Mixin (LM) approach is introduced, which proposes training the main model in an ensemble with the bias-capturing model in order to discourage the main model to encode the information that has been already captured by the bias-capturing model. Furthermore, Rubi [19] suggests a regularization term for adjusting the weighting of logits, thereby reducing the influence of biases during the training process. ReBias [20] is a framework that aims at the independence between the representations of the main model and the bias-capturing model using the Hilbert-Schmidt Independence Criterion [37]. The authors of [21] suggest an approach, namely Learning from Failure (LfF), that uses the generalized Cross-Entropy (CE) loss for training the bias-capturing model to focus on simple samples that are likely to be aligned with the bias and in parallel, the training procedure followed for the main model encourages it to focus on the samples that the bias-capturing model fails to learn, which are expected to be bias-conflicting. The Spread Spurious Attribute (SSA) approach [38] leverages a limited set of bias-labeled samples to train a bias-capturing classifier. Subsequently, the predictions from this classifier are utilized to train a fair model through the minimization of the worst-group loss. In the same direction, the Confidence-based Group Label (CGL) [39] assignment methodology utilizes a protected attribute classifier to assign pseudo-protected attribute labels, while assigning random labels to low confidence samples. Then, CGL is combined with existing state-of-the-art approaches to improve their performance in terms of fairness. An approach for diversifying the bias-conflicting samples through augmentation in the feature space (DistEnt) is proposed in [40]. Moreover, the BiasEnsemble (BE) [41] method proposes an unsupervised technique to discard the bias-conflicting samples from the training dataset. Empirical evidence indicates that this strategy can significantly improve the efficacy of multiple fairness-aware approaches. Finally, the Soft-Contrastive (SoftCon) method [16] constitutes an extension of the BiasCon [16] that makes use of the similarities of the representations extracted by the bias-capturing model in order to enable BiasCon to be employed in scenarios where the protected attribute labels are not available. Although these methods successfully address the lack of protected attribute labels, they exhibit considerably lower performance compared to BL-A methods, as they do not consider appropriate mechanisms for focusing only on the samples that can effectively contribute to the bias mitigation, thus adding noise and preventing the model from learning fairer representations. In contrast, FLAC is a BL-U method that minimizes the MI between the learned model and the protected attributes. This is achieved through the use of a representation derived by a bias-capturing classifier (instead of the attribute labels) and a sampling strategy with optimality guarantees.

c) Mutual Information: MI has been used in various tasks, such as knowledge distillation for preserving the teacher model's knowledge [42], [43], [44], feature selection for selecting the features that are most related to the desired outcome [45], clustering for assessing the clustering quality [46], and generative models for guiding generators in learning different manifolds [47]. LnL [14] proposed using MI in an adversarial framework for bias mitigation, but its performance is considerably lower than most of the bias mitigation approaches, due to the instability that adversarial approaches exhibit. Moreover, [48] introduces a statistical framework that penalizes the MI between the target labels and the protected attribute labels. An effort of converting the Quadratic Mutual Information (QMI) [49] problem to a probability matching problem for the knowledge distillation task is presented in [43]. Inspired by this idea, we propose in this paper a method that leverages the capabilities of MI in order to effectively mitigate the bias, even in extreme data bias scenarios.

III. Methodology

In this section, we introduce the FLAC approach, which is based on correcting bias by focusing on sample representations where bias is identified. As illustrated in Fig. 1, these samples exhibit high (low) similarity for pairs with different (same) target labels and different (same) protected attribute labels, due to the attribute-class associations in the training data. A model demonstrating such behavior can function as a bias-capturing classifier, such as the Vanilla model or a model trained for protected attribute prediction. FLAC utilizes this insight, forcing the main model to showcase the converse behavior, specifically for these certain sample pairs (see Fig. 1). To this end, FLAC (i) suggests a sampling mechanism for the identification of relevant samples and (ii) solves a probability matching problem that encourages the main model to exhibit contrary behavior in terms of pairwise similarities to the bias-capturing model for the chosen sample pairs. Subsequent subsections detail the methodology and provide a theoretical analysis that justifies that minimizing the FLAC loss results in fair representations, that are independent of the protected attributes.

A. Problem Formulation

The problem of mitigating bias in network representations can be formulated as follows. Let $(\mathbf{X}_i, y_i)$ be the i-th training sample of the dataset $\mathcal{D}$, where $\mathbf{X}_i$ is the input image, $y_i \in \mathcal{Y}$ the target, $h(\cdot)$ the model that we are interested to improve in terms of bias, and $b(\cdot)$ the bias-capturing model. Model $h(\cdot)$ is trained on the main task with targets $\mathcal{Y}$, while the bias-capturing model, $b(\cdot)$, is trained to predict protected attributes $t_i \in \mathcal{T}$ towards which the data (and consequently the model) is biased. Note that t_i is not provided by $\mathcal{D}$, thus $b(\cdot)$ is trained on a different dataset (cf. Section III-B). The representations extracted by the penultimate layer of the network are denoted as $\mathbf{h}_i$, while the corresponding bias-capturing classifier representations are denoted as $\mathbf{b}_i$. If the dataset $\mathcal{D}$ consists of samples that exhibit strong dependence between $\mathcal{Y}$ and $\mathcal{T}$, then training the model $h(\cdot)$ using only the task-specific loss function (e.g. CE for the multi-class classification task or a contrastive loss) will introduce bias into the model since it will rely on features that encapsulate

TABLE I
SCENARIOS WHERE FLAC VARIATIONS CAN BE APPLIED COMPARED TO OTHER BL-A AND BL-U APPROACHES

method	bias labels unavailable		non-categorical bias
	$\mathcal{D}$	$\mathcal{D}'$	
BL-A	✗	n.a.	✗
BL-U	✓	✓	✓
FLAC	✓	✗	✗
FLAC-B	✓	✓	✓

information about t_i for predicting y_i. Therefore, the goal is to eliminate the dependencies between representations $\mathbf{h}$ and $\mathbf{b}$.

B. Bias-Capturing Classifier

In many real-world scenarios, the attributes introducing bias are not categorical or the corresponding labels are not available. To address this issue, we make use of a bias-capturing classifier, as in [16], [17], [20]. This acts as a feature extractor that encapsulates information related to the attributes that introduce bias to $h(\cdot)$.

If the protected attributes are categorical, the $b(\cdot)$ is trained on a dataset $\mathcal{D}' = \{(\mathbf{X}'_i, t_i)\}$ in which the protected attribute $t_i \in \mathcal{T}$ is provided (unlike $\mathcal{D}$). For instance, if *gender* is the attribute related to bias, the bias-capturing classifier is trained to predict genders. In some cases, the model is biased towards attributes that are not categorical (e.g., texture or background) or there are no available datasets providing protected attribute annotations to train a bias-capturing model. Following previous literature [16], [17], [20], we opt for models trained on $\mathcal{D}$ that predict the task-specific targets $\mathcal{Y}$. For these networks, the penultimate layer is expected to adequately capture the attributes of interest [20]. To maintain clarity, when a Vanilla model is employed as a bias-capturing model, we label it as FLAC-B. Table I outlines the applicability characteristics of FLAC variations in comparison to other BL-A and BL-U methods.

C. FLAC

Discouraging the main model from learning to predict the protected attribute labels can be achieved by minimizing $\mathcal{I}(\mathbf{h}_i, t_i)$, where $\mathcal{I}(\cdot,\cdot)$ denotes MI. However, this is a task of high complexity and additionally it would require access to the protected attribute labels $\mathcal{T}$. Instead, we propose to use the pairwise similarity between the model representation and the representation derived by a bias-capturing classifier, i.e., a predictor of t_i, to minimize $\mathcal{I}(\mathbf{h}_i, t_i)$. More specifically, the target is to match the probability distributions of distances derived by the bias-capturing classifier features with the distributions of similarities derived by the representations of the main model for a certain subset of sample pairs. By doing so, we bring the samples with different protected attribute labels (i.e., $t_i \neq t_j$) and the same target label (i.e., $y_i = y_j$) closer to each other while increasing the distance between samples with the same protected attribute label (i.e., $t_i = t_j$) and different target labels (i.e., $y_i \neq y_j$). For all possible pairs of batch samples' indices (i, j) we define the following sets:

$$\mathcal{A}_{10} = \{(i,j) \mid y_i = y_j \wedge t_i \neq t_j\}$$
$$\mathcal{A}_{01} = \{(i,j) \mid y_i \neq y_j \wedge t_i = t_j\}$$
$$\mathcal{A}_{11} = \{(i,j) \mid y_i = y_j \wedge t_i = t_j, i \neq j\}$$
$$\mathcal{A}_{00} = \{(i,j) \mid y_i \neq y_j \wedge t_i \neq t_j\}$$
$$\mathcal{A} = \mathcal{A}_{10} \cup \mathcal{A}_{01} \cup \mathcal{A}_{11} \cup \mathcal{A}_{00}$$

Then, involved pairs of samples should satisfy the following condition:

$$(y_i = y_j \wedge t_i \neq t_j) \vee (y_i \neq y_j \wedge t_i = t_j) \tag{1}$$

forming the set $\mathcal{S} = \mathcal{A}_{10} \cup \mathcal{A}_{01}$. Note that involving all the possible sample pairs could have an adverse effect on the main model as it would lead to reducing (increasing) the similarity of samples with the same (different) target labels (see Section III-D). However, $\mathcal{T}$ labels are not available, thus a criterion needs to be defined for inferring whether two samples share the same protected attribute label. Let $K(\cdot)$ be a kernel function and $\mathcal{A}$ the set of all the possible pairs of samples, then the protected attribute equality is determined by:

$$K(\mathbf{b}_i, \mathbf{b}_j) > \frac{\max\limits_{u,v \in \mathcal{A}} K(\mathbf{b}_u, \mathbf{b}_v) + \min\limits_{u,v \in \mathcal{A}} K(\mathbf{b}_u, \mathbf{b}_v)}{2} \Rightarrow$$
$$\Rightarrow t_i = t_j,\ (i,j) \in \mathcal{A} \tag{2}$$

otherwise $t_i \neq t_j$. It should be noted that the proposed methodology will not be negatively impacted even if $K(\mathbf{b}_i, \mathbf{b}_j)$ falls below or exceeds this threshold for (i, j) with $t_i = t_j$ or $t_i \neq t_j$, respectively. This is because the objective remains to enhance the similarity of (i, j) where $y_i = y_j$ or reduce it when $y_i \neq y_j$.

Then, the task-specific network's, $h(\cdot)$, probability distributions of the pairwise similarities can be defined as follows:

$$p^{(h)}_{i|j} = \frac{K(\mathbf{h}_i, \mathbf{h}_j)}{\sum_{k:(k,j)\in\mathcal{S}} K(\mathbf{h}_k, \mathbf{h}_j)} \in [0,1],\ (i,j) \in \mathcal{S}. \tag{3}$$

Accordingly, for the bias-capturing model, we calculate the probability distributions of dissimilarities (i.e., $(1 - K(\cdot))$):

$$p^{(b)}_{i|j} = \frac{1 - K(\mathbf{b}_i, \mathbf{b}_j)}{\sum_{k:(k,j)\in\mathcal{S}} 1 - K(\mathbf{b}_i, \mathbf{b}_j)} \in [0,1],\ (i,j) \in \mathcal{S}. \tag{4}$$

As regards the similarity kernel function, in this work we opted for the student's t kernel that demonstrates good performance on classification tasks [42]:

$$K(\mathbf{a}, \mathbf{b}) = \frac{1}{1 + ||\mathbf{a} - \mathbf{b}||_2}. \tag{5}$$

Having calculated the probability distributions of (3) and (4) for the pairs of samples in $\mathcal{S}$, our goal is to train a model that demonstrates high (low) similarity for the sample pairs that the bias-capturing classifier exhibit low (high) similarity. The divergence between these distributions can be calculated using Jeffreys divergence [50], a symmetric version of Kullback-Leibler divergence:

$$\mathcal{L}_{FLAC} = \sum_{(i,j)\in\mathcal{S}} \left(p^{(b)}_{i|j} - p^{(h)}_{i|j}\right) \cdot \left(\log p^{(b)}_{i|j} - \log p^{(h)}_{i|j}\right). \tag{6}$$

Then, the final loss can be defined as follows:

$$\mathcal{L} = \mathcal{L}_{task} + \alpha \cdot \mathcal{L}_{FLAC}, \tag{7}$$

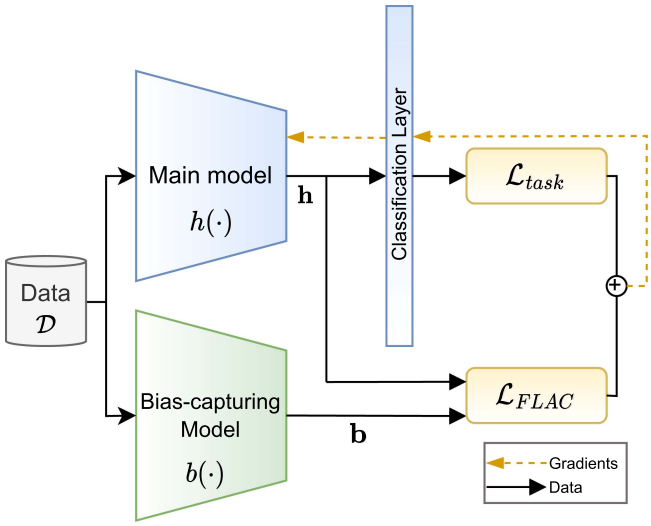


Fig. 2. Illustration of the proposed framework.

where α is a hyperparameter and $\mathcal{L}_{task}$ is the cost function of the task (e.g., the CE or a contrastive loss). Note that FLAC fits better with pairwise losses as $\mathcal{L}_{task}$, due to its dependency on pairwise similarities. Fig. 2 presents the proposed framework.

D. Theoretical Analysis

Let us denote with $\mathcal{B} = \{\mathbf{X}_i\}_{i=1}^{N}$ a batch with N samples. Following the notation of Section III-C, we additionally consider the set Ω as a placeholder variable that upon declaration equals either $\mathcal{S}$ or $\mathcal{A}$. Finally, we assume that for a well-trained bias capturing classifier $\exists \epsilon > 0$, such that $\forall (i,j) \in \Omega$:

$$t_i \neq t_j \Rightarrow K(\mathbf{b}_i, \mathbf{b}_j) \in [0, \epsilon) \tag{8}$$

$$t_i = t_j \Rightarrow K(\mathbf{b}_i, \mathbf{b}_j) \in (1-\epsilon, 1] \tag{9}$$

When the contrastive loss, denoted by $\mathcal{L}_{task}$, is minimized after training, then $\exists \delta_{10}, \delta_{01}, \delta_{11}, \delta_{00}, \epsilon' > 0$ with $\epsilon' \leq \delta_{10}, \delta_{01}, \delta_{11}, \delta_{00}$, such that:[1]

$$K(\mathbf{h}_i, \mathbf{h}_j) \in \begin{cases} (1-\delta_{10}-\epsilon', 1-\delta_{10}+\epsilon'], \text{if(i,j)} \in \mathcal{A}_{10} \\ [\delta_{01}-\epsilon', \delta_{01}+\epsilon'), \text{if(i,j)} \in \mathcal{A}_{01} \\ (1-\delta_{11}-\epsilon', 1-\delta_{11}+\epsilon'], \text{if(i,j)} \in \mathcal{A}_{11} \\ [\delta_{00}-\epsilon', \delta_{00}+\epsilon'), \text{if(i,j)} \in \mathcal{A}_{00} \end{cases} \tag{10}$$

However, in cases of existing bias in which the model has learned unwanted shortcuts (i.e., $\mathbf{h}_i \not\perp t_i$), the magnitude of δ_{10}, δ_{01} monotonically depends on the amount of shared information between $\mathbf{h}_i$ and t_i,[2] while the magnitude of δ_{11}, δ_{00} not. This indicates the need to control δ_{10}, δ_{01} with an additional loss term that acts at least on $(i,j) \in \mathcal{S}$.[3] To this end, in this work we propose the consideration of an additional loss term, namely $\mathcal{L}_{FLAC}$, defined by (6), which will be minimized jointly with $\mathcal{L}_{task}$. This, then implies the following:

$$\begin{aligned} &\mathcal{L}_{FLAC} = 0 \Rightarrow \\ &\Rightarrow \left(p_{i|j}^{(b)} - p_{i|j}^{(h)} = 0\right) \vee \left(\log p_{i|j}^{(b)} - \log p_{i|j}^{(h)} = 0\right) \Rightarrow \\ &\Rightarrow p_{i|j}^{(b)} = p_{i|j}^{(h)} \Rightarrow \\ &\Rightarrow \frac{K(\mathbf{h}_i, \mathbf{h}_j)}{\sum_{k:(k,j)\in\Omega} K(\mathbf{h}_k, \mathbf{h}_j)} = \frac{1 - K(\mathbf{b}_i, \mathbf{b}_j)}{\sum_{k:(k,j)\in\Omega} 1 - K(\mathbf{b}_k, \mathbf{b}_j)} \Rightarrow \\ &\Rightarrow K(\mathbf{h}_i, \mathbf{h}_j) = \alpha(j) \cdot (1 - K(\mathbf{b}_i, \mathbf{b}_j)) \end{aligned} \tag{11}$$

where $(i,j) \in \Omega$. Considering for simplicity $\epsilon \approx \epsilon' \approx \delta_{11} \approx \delta_{00} \approx 0$ and $\delta_{10} \approx \delta_{01} \overset{\text{def}}{=} \delta$, based on (10) we get:

$$\begin{aligned} \alpha(j) &= \frac{\sum_{k:(k,j)\in\Omega} K(\mathbf{h}_k, \mathbf{h}_j)}{\sum_{k:(k,j)\in\Omega} 1 - K(\mathbf{b}_k, \mathbf{b}_j)} = \\ &= \frac{\sum_{k:(k,j)\in\mathcal{A}_{10}\cup\mathcal{A}_{01}\cup\mathcal{A}_{11}} K(\mathbf{h}_k, \mathbf{h}_j)}{\sum_{k:(k,j)\in\mathcal{A}_{10}\cup\mathcal{A}_{00}} 1 - K(\mathbf{b}_k, \mathbf{b}_j)} = \\ &= \frac{(1-\delta)\cdot \mid \mathcal{A}_{10}^j \mid + \delta \cdot \mid \mathcal{A}_{01}^j \mid + \mid \mathcal{A}_{11}^j \mid}{\mid \mathcal{A}_{10}^j \mid + \mid \mathcal{A}_{00}^j \mid} = \\ &= \frac{\mid \mathcal{A}_{10}^j \mid + \mid \mathcal{A}_{11}^j \mid + \delta \cdot \left(\mid \mathcal{A}_{01}^j \mid - \mid \mathcal{A}_{10}^j \mid \right)}{\mid \mathcal{A}_{10}^j \mid + \mid \mathcal{A}_{00}^j \mid} \end{aligned} \tag{12}$$

where $\mathcal{A}_{uv}^j = \{k \mid (k,j) \in \mathcal{A}_{uv}\}$ and $u, v \in \{0,1\}$.

Additionally, based on the (8), (9) and (11), minimizing (6) results in:

$$K(\mathbf{h}_i, \mathbf{h}_j) = \begin{cases} \alpha(j) & \text{, if } (i,j) \in \mathcal{A}_{10} \\ 0 & \text{, if } (i,j) \in \mathcal{A}_{01} \\ 0 & \text{, if } (i,j) \in \mathcal{A}_{11} \\ \alpha(j) & \text{, if } (i,j) \in \mathcal{A}_{00} \end{cases} \tag{13}$$

So, if $\Omega = \mathcal{S}$, then $\mid \mathcal{A}_{11}^j \mid = \mid \mathcal{A}_{00}^j \mid = 0$ implying $\alpha(j) = 1$ (cf. (12)),[4] which results in an ideal form of (13), namely:

$$K(\mathbf{h}_i, \mathbf{h}_j) = \begin{cases} 1 & \text{, if } (i,j) \in \mathcal{A}_{10} \\ 0 & \text{, if } (i,j) \in \mathcal{A}_{01} \end{cases} \tag{14}$$

accomplishing the very small and independent δ_{10}, δ_{01} existence. In contrast, if $\Omega = \mathcal{A}$, then:

$$\mid \mathcal{A}_{11}^j \mid \geq 0$$

$$\mid \mathcal{A}_{00}^j \mid \geq 0$$

so depending on the sizes of the above sets $\alpha(j)$ can greatly vary. Moreover, not only the 3rd branch of (13) forces pairs

[1] It has been shown that optimizing the standard cross-entropy loss: (i) corresponds to an approximate bound-optimizer of an underlying pairwise loss [51] and (ii) produces a feature space with inter-class distances being greater than intra-class ones [52]. Nevertheless, accomplishing low $\delta_{10}, \delta_{01}, \delta_{11}, \delta_{00}$ applying to all i, j is hardly possible. Hence, adopting the cross-entropy will only approximate the optimality derived by the contrastive loss.

[2] If the shared information level is low, then there exist very small δ_{10}, δ_{01}. If the shared information level is high, then the δ_{10}, δ_{01} can indeed be high but the cardinality of $\mathcal{A}_{10}$ and $\mathcal{A}_{01}$ is very small, much smaller than the cardinality of $\mathcal{A}_{11}$ and $\mathcal{A}_{00}$ (cf. Section VI & Fig. 5).

[3] An analysis showing that acting only on $\mathcal{S}$ is the best option in comparison to acting on the whole space $\mathcal{A}$, is considered below.

[4] Note that when bias is uniformly distributed across classes $\mid \mathcal{A}_{01}^j \mid = \mid \mathcal{A}_{10}^j \mid$, which is the case in all experiments of this work. We consider exploring atypical cases of non-uniform bias distribution across classes as out of this works scope and leave it for future work.

$(i,j) \in \mathcal{A}_{11}$ to have dissimilar embeddings $\mathbf{h}_i$ and $\mathbf{h}_j$ -thus lowering the accuracy-, but also the training becomes highly unstable with each batch imposing different $\alpha(j)$ (and consequently $K(\mathbf{h}_i, \mathbf{h}_j)$) values. Thus, we opt for considering $\mathcal{L}_{FLAC}$ acting on $\Omega = \mathcal{S}$ to address bias and achieve higher accuracy levels.

By doing so, we argue that minimizing the total loss (i.e., (7)) minimizes mutual information between the main model's representations $\mathbf{h}$ and the corresponding protected attributes t. In order to prove this, we consider the definition of QMI as defined in [49], namely (15):

$$\mathcal{I}(\mathbf{h},t) = \sum_t \int_{\mathbf{h}} p(\mathbf{h},t)^2 d\mathbf{h} + \sum_t \int_{\mathbf{h}} (p(\mathbf{h})P(t))^2 d\mathbf{h} - \\ - 2\sum_t \int_{\mathbf{h}} p(\mathbf{h},t)p(\mathbf{h})P(t)d\mathbf{h}, \tag{15}$$

whose 3 terms are called information potentials and are denoted by V_{IN}, V_{ALL}, and V_{BTW}, respectively. Then, the information potentials can be estimated as follows:

$$V_{IN} = \frac{1}{N^2}\sum_{p=1}^{N_c}\sum_{k=1}^{J_p}\sum_{l=1}^{J_p} K(\mathbf{h}_{pk}, \mathbf{h}_{pl}), \tag{16}$$

$$V_{ALL} = \frac{1}{N^2}\sum_{p=1}^{N_c}\left(\frac{J_p}{N}\right)^2 \sum_{k=1}^{N}\sum_{l=1}^{N} K(\mathbf{h}_k, \mathbf{h}_l), \tag{17}$$

$$V_{BTW} = \frac{1}{N^2}\sum_{p=1}^{N_c}\frac{J_p}{N}\sum_{j=1}^{J_p}\sum_{k=1}^{N} K(\mathbf{h}_{pj}, \mathbf{h}_k), \tag{18}$$

where N, N_c, and J_p are the number of samples, classes, and samples belonging to the class p, respectively. Note that J_p is equal to $\frac{N}{N_c}$ as QMI should be calculated on balanced data. V_{IN} consists of the interactions between pairs within each protected attribute class, V_{ALL} consists of the interactions between all pairs, and V_{BTW} consists of interactions between samples of each protected attribute class against all samples. Considering the optimal scenario with $\mathcal{L}_{task} = 0$ and $\mathcal{L}_{FLAC} = 0$, the (10) holds for very small δ_{10}, δ_{10}, δ_{11}, δ_{00} (for simplicity we consider $\delta_{10} \approx \delta_{10} \approx \delta_{11} \approx \delta_{00} \approx \epsilon' \approx 0$) and the information potentials, V_{IN}, V_{ALL}, and V_{BTW}, result in the following:

$$V_{IN} = \frac{1}{N^2} \mid \mathcal{A}_{11} \mid, \tag{19}$$

$$V_{ALL} = \frac{1}{N^2}\sum_{p=1}^{N_c}\left(\frac{J_p}{N}\right)^2 (\mid \mathcal{A}_{11} \mid + \mid \mathcal{A}_{10} \mid) = \\ \overset{J_p=\frac{N}{N_c}}{=} \frac{1}{N^2}\frac{1}{N_c}(\mid \mathcal{A}_{11} \mid + \mid \mathcal{A}_{10} \mid), \tag{20}$$

$$V_{BTW} = \frac{1}{N^2}\sum_{p=1}^{N_c}\frac{J_p}{N}\frac{\mid \mathcal{A}_{11} \mid + \mid \mathcal{A}_{10} \mid}{N_c} = \\ \overset{J_p=\frac{N}{N_c}}{=} \frac{1}{N^2}\frac{1}{N_c}(\mid \mathcal{A}_{11} \mid + \mid \mathcal{A}_{10} \mid). \tag{21}$$

Finally, given that $\mid \mathcal{A}_{10} \mid = (N_c - 1)\cdot \mid \mathcal{A}_{11} \mid$, in the optimal scenario of $\mathcal{L} = 0$ the QMI between the representations $\mathbf{h}$ and the corresponding protected attributes t, is calculated as:

$$\begin{aligned}\mathcal{I}(\mathbf{h},t) &= V_{IN} + V_{ALL} - 2V_{BTW} = V_{IN} - V_{BTW} = \\ &= \frac{1}{N^2}\left(\mid \mathcal{A}_{11} \mid - \frac{\mid \mathcal{A}_{11} \mid + \mid \mathcal{A}_{10} \mid}{N_c}\right) = \\ &= \frac{1}{N^2}\left(\mid \mathcal{A}_{11} \mid - \frac{\mid \mathcal{A}_{11} \mid + (N_c-1)\mid \mathcal{A}_{11} \mid}{N_c}\right) = \\ &= \frac{1}{N^2}\left(\mid \mathcal{A}_{11} \mid - \frac{\mid \mathcal{A}_{11} \mid}{N_c} - \frac{(N_c-1)\mid \mathcal{A}_{11} \mid}{N_c}\right) = \\ &= \frac{1}{N^2}\left(\left(1 - \frac{1}{N_c}\right) \mid \mathcal{A}_{11} \mid - \frac{(N_c-1)}{N_c} \mid \mathcal{A}_{11} \mid\right) = \\ &= 0\end{aligned} \tag{22}$$

Thus, FLAC can indeed lead to fair representations, that are independent of the protected attributes.

IV. Experimental Setup

A. Datasets

a) Datasets with Artificially Injected Bias: Biased-MNIST constitutes a biased version of the original MNIST dataset [53], introduced in [20] as a standard benchmark for evaluating the effectiveness of bias mitigation methods. It consists of digits with colored backgrounds (10 colors in total). The bias is introduced through the association of each digit with a certain color. The probability of a digit having biased background is denoted as q, while a random background color is assigned with probability $1 - q$. This way, higher q values lead to a stronger association between digits and colors and thus more biased data. Following previous works using this dataset, here we consider four variations of the Biased-MNIST w.r.t. values of q, namely 0.99, 0.995, 0.997, and 0.999.

b) Established computer vision datasets: We also evaluate the proposed approach on three established computer vision datasets, namely CelebA [22], UTK-Face [23], and 9-Class ImageNet [20]. As regards the CelebA dataset that consists of more than 200,000 facial images annotated with 40 binary attributes, the *gender* is the protected attribute, while the *HeavyMakeup* and the *BlondHair* constitute the two target labels, as in [16]. For the UTKFace, which is a dataset consisting of over 20,000 facial images with *gender*, *race*, and *age* labels, we consider the *gender* as the target label, while *race* and *age* as the protected attributes. For UTKFace, we enforce a dependence of 90% between the target and the protected attribute following the experimental setup of [16]. The Corrupted-Cifar10 [24] dataset consists of 10 classes with texture biases evident in the training data and it offers four correlation ratios: 0.95, 0.98, 0.99, and 0.995. The 9-Class ImageNet, a subset of the ImageNet dataset [54] consisting of 9 super-classes, is employed to evaluate the proposed method in settings where bias is not explicitly associated with a given attribute (e.g., texture bias). In addition, the ImageNet-A [55], which comprises ImageNet samples that Vanilla models often fail to classify, is also involved in the experiments as a test set.

B. Model Architectures

For the controlled experiments conducted for Biased-MNIST, we employ the CNN architecture proposed in [20], namely *SimpleConvNet* that consists of four convolutional layers with 7×7 kernels and a fully connected layer with 128 neurons. For the CelebA, UTKFace, Corrupted-Cifar10, and 9-Class ImageNet, we employ the Resnet-18 [56] architecture for both the main and the bias-capturing models, except for the 9-Class ImageNet, where we use the BagNet-18 [57] as the bias-capturing model following the practice of [16], [17], [20]. BagNets mainly consist of 1×1 convolutions and they are found to be more prone to texture bias [20]. Note that all model selections are based on the previous literature to ensure comparability.

C. Baseline Methods

The proposed method is compared to 9 state-of-the-art approaches. Of those, BL-U approaches include the LM [18], Rubi [19], ReBias [20], LfF [21], and SoftCon [16]. The BL-A approaches considered in our comparative study are the LNL [14], EnD [15], DistEnt [40], BE [41], BC-BB [16], and FairKL [17]. More details about the competing methods can be found in Section II. For ensuring a fair comparison between the approaches, it should be highlighted that:

- FLAC combined with a Vanilla biased model as a bias-capturing model (i.e., FLAC-B) allows for a direct comparison with BL-U approaches, devoid of any reliance on protected attribute labels.
- Conversely, FLAC combined with a bias-capturing model trained using attribute labels of $\mathcal{D}'$ holds an advantage over BL-U methods that do not use any extra data and a disadvantage compared to BL-A methods due to the error of the bias-capturing models in predicting the protected attributes.

D. Implementation Details and Evaluation Protocol

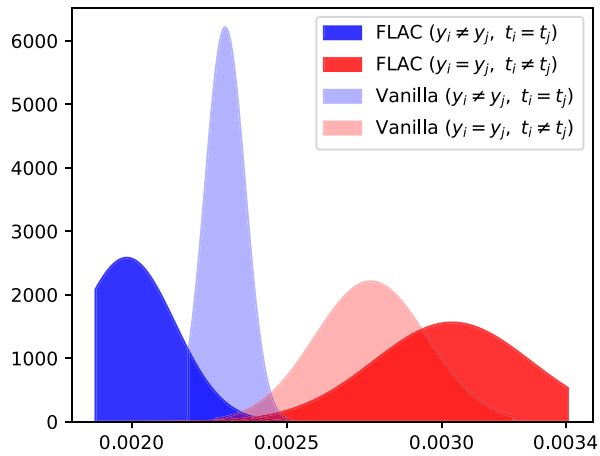
We employ the Adam optimizer for all the experiments with an initial learning rate of 0.001 that decays by a factor of 0.1 at 1/3 and 2/3 of the total training epochs, the weight decay is equal to 10^{-4} and the batch size is set to 128, unless stated otherwise. For the Biased-MNIST, models are trained for 80 epochs. The values of α for each BiasedMNIST variant, 0.99, 0.995, 0.997, and 0.999 are 110, 1500, 2000, and 10,000 respectively. Note that such large values of α were selected due to the large discrepancy between the values of $\mathcal{L}_{FLAC}$ and $\mathcal{L}_{task}$. The α values were chosen following an initial grid search. There is a possibility that a more comprehensive hyperparameter tuning process could lead to improved outcomes. Additionally, it has been noticed that small α can lead to unstable training behavior. The augmentations applied for the established computer vision datasets (i.e., CelebA, UTKFace, and 9-Class ImageNet) are the *random resized crop* and *random horizontal flip*. For the CelebA, models are trained for 40 epochs with α equal to 20,000 for the *HeavyMakeup* classification task, while 10 training epochs with α equal to 30,000 are used for the *BlondHair* classification task. The images are resized to 224×224. For UTKFace, models are trained for 20 epochs, α is set to 1,000 and image size to 64×64. For the experiments leveraging bias-capturing models trained with the protected attributes as targets, we utilized the FairFace dataset which provides *gender*, *race*, and *age* annotations. Specifically, the models for *race*, *age*, and *gender* demonstrate 89.86%, 98.80%, and 95.79% accuracy on UTKFace and CelebA, respectively. For Corrupted-Cifar10, the models are trained for 100 epochs using a cosine annealing scheduler, α is equal to 100,000, and image size is 32×32. For 9-Class ImageNet, the cosine annealing learning rate scheduler is used, the number of training epochs is equal to 250, α is set to 1,000, and image size to 224×224. As $\mathcal{L}_{task}$ we consider both the Supervised Contrastive (SupCon) [58] and the CE loss across all the experiments in order to provide a fair comparison between the competing bias mitigating methods. In addition, only subtle performance discrepancies have been noticed when assigning a weight to this loss. All the experiments were conducted on a single NVIDIA RTX-3090 GPU and repeated for 5 different random seeds (all results reported in Section V refer to the corresponding mean scores).

The test set used for BiasedMNIST evaluation has $q = 0.1$ in order to be unbiased (i.e., no association between digits and colors). For CelebA and UTKFace, we use the unbiased and bias-conflict test sets provided by [16] for the evaluation. For Corrupted-Cifar10, the official unbiased test set is considered. Finally, for the 9-Class ImageNet experiments, two test sets are involved, (a) the official 9-Class ImageNet test set and (b) the ImageNet-A. For all the above, accuracy is employed as an evaluation metric. In addition, we measure the unbiased accuracy on 9-Class ImageNet test set by using the texture bias annotations provided by [20] and averaging the accuracy calculated for each one of the texture groups. Furthermore, apart from the typical performance metric (i.e, accuracy) we also report the performance of FLAC in terms of three fairness metrics for disparate impact and disparate mistreatment, namely *p% rule* [59], Difference in False Positive Rates (DFPR), and Difference in False Negative Rates (DFNR) [60], [61].

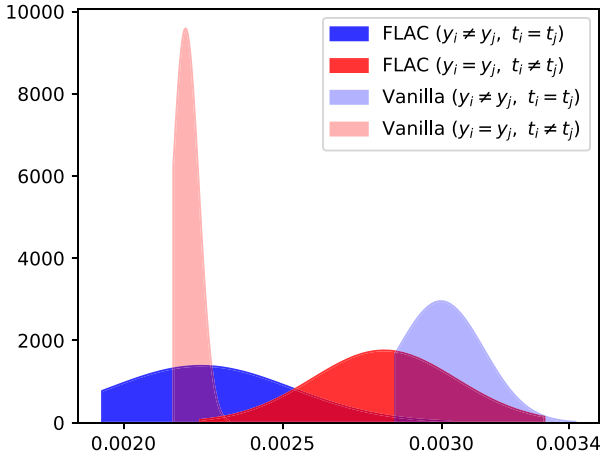
V. Results

A. Controlled Experiments

Table II presents the performance comparison of the proposed approach against 5 BL-U and 4 BL-A methods. *Vanilla* refers to the model performance using only the CE loss without considering any bias mitigation algorithm. According to Table II, the proposed method consistently outperforms all competing methods, even the BL-A ones. In particular, FLAC enhances the accuracy by 0.1%-0.6% for the different q values compared to the best performing BL-A method and by 3.5%-29.1% compared to the best performing BL-U approach. Furthermore, it is worth noting that while q increases, FLAC is capable of maintaining very high accuracy, which is not the case for most of the compared methods. For instance, for the extreme q value of 0.999, FLAC achieves 94.1% accuracy, while the best-performing BL-U method achieves 65%. In addition, note that FLAC combined with CE loss significantly outperforms the corresponding competing methods that use CE as $\mathcal{L}_{task}$. As



(a) Distributions of Equation (3) for Vanilla and FLAC trained on Biased-MNIST with $q = 0.99$ and achieves 88.9% and 96.9% accuracy, respectively. Using FLAC, the similarity between samples with the same or different label is increased or decreased respectively, compared to Vanilla.



(b) Distributions of Equation (3) for Vanilla (11.8% acc.) and FLAC (89.3% acc.) trained on Biased-MNIST with $q = 0.999$. In this evaluation setup, Vanilla results in an extremely biased model that exhibits high (low) similarities for samples with the same (different) protected attribute label and different (same) target. Here, the FLAC's impact is significant as it manages to flip the distributions.

Fig. 3. The distributions of similarities or/and distances for the samples that satisfy (1). *Red* color represents the sample pairs with the same target, but different protected attribute label, while *blue* depicts the pairs with different target, but the same protected attribute labels.

TABLE II
EVALUATION ON BIASED-MNIST FOR DIFFERENT BIAS LEVELS

methods	L_{task}	BL-U	q			
			0.99	0.995	0.997	0.999
Vanilla	CE	✓	90.8±0.3	79.5±0.1	62.5±2.9	11.8±0.7
LNL	CE	✗	86.0±0.2	72.5±0.9	57.2±2.2	18.2±1.2
EnD	CE	✗	94.8±0.3	94.0±0.6	82.7±0.3	59.5±2.3
BC-BB	Con.*	✗	98.1±0.1	97.7±0.1	97.3±0.1	94.0±0.6
FairKL	CE	✗	95.9±0.2	94.8±0.5	93.9±1.1	79.9±4.3
FairKL	Con.	✗	97.9±0.0	97.0±0.0	96.2±0.2	90.5±1.5
LM	CE	✓	91.5±0.4	80.9±0.9	56.0±4.3	10.5±0.6
Rubi	CE	✓	85.9±0.1	71.8±0.5	49.6±1.5	10.6±0.5
ReBias	CE	✓	88.4±0.6	75.4±1.0	65.8±0.3	26.5±1.4
LfF	CE	✓	95.1±0.1	90.3±1.4	63.7±20.3	15.3±2.9
DisEnt[5]	CE	✓	96.1±0.4	91.5±0.3	81.9±6.3	38.3±20.1
DisEnt + BE[5]	CE	✓	94.0±1.7	90.8±3.0	80.9±11.2	26.4±2.4
SoftCon	Con.	✓	95.2±0.4	93.1±0.2	88.6±1.0	65.0±3.2
FLAC	CE	✓	96.9±0.0	94.9±0.5	94.1±0.5	89.3±1.3
FLAC	Con.	✓	97.9±0.1	96.8±0.0	95.8±0.2	89.4±0.8
FLAC	Con.*	✓	**98.7**±0.0	**98.2**±0.1	**97.8**±0.1	**94.1**±1.1

Underlined and dotted-underlined values refer to the second best performing BL-U and the best performing BL-A methods, respectively. Con. Refer to contrastive losses. * refers to the performance with color-jittering (i.e., randomly changing the brightness, contrast, saturation, and hue of images).

regards the selection of $\mathcal{L}_{task}$, as expected opting for a pairwise loss (i.e., SupCon) results in improved accuracy as justified in Section III-D.

As shown in Fig. 3(a), which presents the distributions (with $q = 0.99$) derived by (3) for Vanilla and FLAC the proposed method increases (decreases) the similarities of samples with the same (different) target labels and different (same) protected attribute labels compared to Vanilla. Furthermore, for $q = 0.999$, where Vanilla results in a model that learned the background colors instead of the digits, FLAC is capable of learning the proper distributions as depicted in Fig. 3(b).

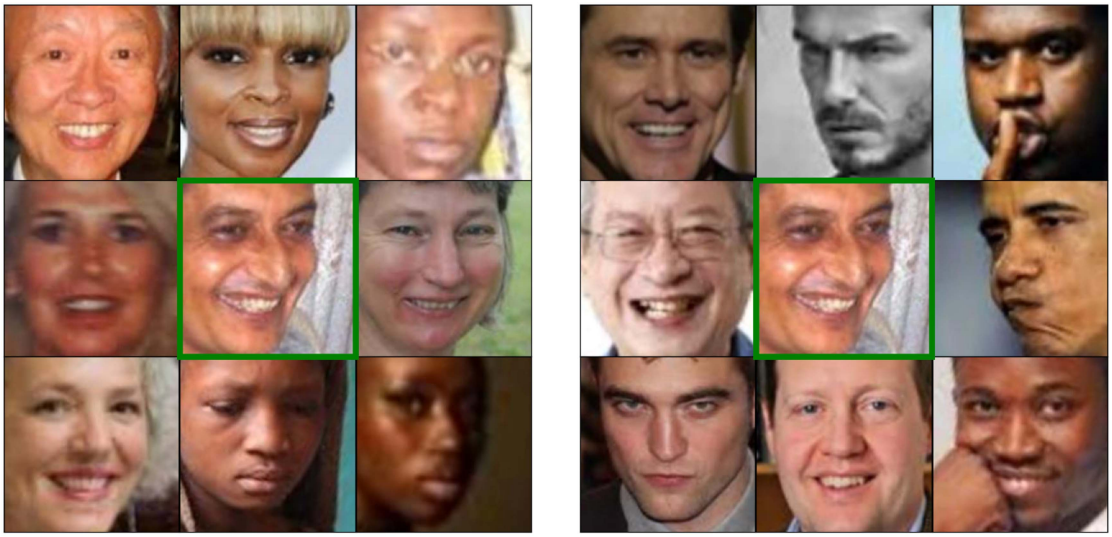
TABLE III
EVALUATION OF THE PROPOSED METHOD ON CELEBA FOR TWO DIFFERENT TARGET ATTRIBUTES, NAMELY *HEAVYMAKEUP* AND *BLONDHAIR*, WITH *GENDER* AS THE PROTECTED ATTRIBUTE

methods	L_{task}	BL-U	target			
			BlondHair		HeavyMakeup	
			unbiased	bias-conflict	unbiased	bias-conflict
Vanilla	CE	✓	79.0±0.1	59.0±0.1	76.0±0.8	55.2±1.9
LNL	CE	✗	80.1±0.8	61.2±1.5	76.4±2.3	57.2±4.6
DI	CE	✗	90.9±0.3	86.3±0.4	74.3±1.1	53.8±1.6
EnD	CE	✗	86.9±1.0	76.4±1.9	74.8±1.8	53.3±3.6
BC-BB	Con.	✗	91.4±0.0	87.2±0.2	78.6±1.8	63.5±3.7
FairKL[5]	Con.	✗	81.7±1.7	69.9±2.4	77.4±1.1	57.2±1.6
LfF	CE	✓	84.2±0.3	81.2±1.4	66.2±1.2	45.5±4.3
SoftCon	Con.	✓	84.1	74.4	77.4	61.0
FLAC	CE	✓	90.1±0.3	87.6±0.6	**85.4**±1.9	79.1±4.1
FLAC	Con.	✓	**91.2**±0.3	**88.7**±0.5	84.7±1.7	78.8±4.6
FLAC-B	Con.	✓	87.0±0.6	84.9±2.2	82.6±3.0	**79.4**±4.1

B. Evaluation on Established Datasets

Table III compares[5] the methods for the two tasks of CelebA, namely predicting *BlondHair* and *HeavyMakeup*. The *unbiased* and the *bias-conflict* refer to test sets that have balance bias-aligned and bias-conflict samples and only bias-conflict samples, respectively. As regards the *BlondHair* prediction task, the proposed method outperforms the best competing BL-U method by 7% and 7.5% on the *unbiased* and the *bias-conflict* test sets, respectively. In addition, FLAC manages to outperform even the best competing BL-A method by 1.5% on the *bias-conflict* test set, while demonstrating competitive performance on the *unbiased* test set (i.e., -0.2%). It should be stressed that FLAC attains this competitive performance despite the bias-capturing model's errors, unlike BL-A methods which directly utilize bias labels. As regards the most challenging task, namely *HeavyMakeup*, the

[5]Reimplemented based on the code provided by the authors.



(a) Method: Vanilla, Query image: non-white male (b) Method: FLAC, Query image: non-white male

Fig. 4. The top 8 images that are most similar to queries representing a minority group of UTKFace (i.e., non-white males) based on the representations derived by Vanilla and FLAC approaches. Images with green borders denote the query samples.

TABLE IV
EVALUATION OF THE PROPOSED METHOD ON UTKFACE FOR 2 DIFFERENT PROTECTED ATTRIBUTES, NAMELY *RACE* AND *AGE*, WITH *GENDER* AS THE TARGET ATTRIBUTE

methods	L_{task}	BL-U	bias			
			race		age	
			unbiased	bias-conflict	unbiased	bias-conflict
Vanilla	CE	✓	87.4±0.3	79.1±0.3	72.3±0.3	46.5±0.2
LNL	CE	✗	87.3±0.3	78.8±0.6	72.9±0.1	47.0±0.1
DI	CE	✗	88.9±1.2	89.1±1.6	75.6±0.8	60.0±0.2
EnD	CE	✗	88.4±0.3	81.6±0.3	73.2±0.3	47.9±0.6
BC-BB	Con.	✗	91.0±0.2	89.2±0.1	79.1±0.3	71.7±0.8
FairKL[6]	Con.	✗	85.5±0.7	80.4±1.0	72.7±0.2	48.6±0.6
SoftCon	Con.	✓	87.0	80.2	74.6	59.2
FLAC	CE	✓	90.9±0.3	90.2±0.4	78.7±1.3	**81.1**±1.6
FLAC	Con.	✓	**92.0**±0.2	92.2±0.7	80.6±0.7	71.6±2.6
FLAC-B	Con.	✓	91.4±0.2	**93.2**±0.3	**80.7**±0.7	77.0±3.7

proposed method significantly outperforms both the BL-A and the BL-U approaches. In particular, FLAC achieves 85.4% and 79.1% accuracy on the *unbiased* and the *bias-conflict* test sets, while the best performing BL-A method only manages to get 78.6% and 63.5% in terms of accuracy, respectively. Here, FLAC with CE demonstrates higher performance compared to FLAC combined with SupCon loss. Finally, the FLAC's performance when employing the Vanilla model as a bias capturing classifier is reported as FLAC-B in Table III. It can be easily noticed that FLAC-B demonstrates significantly improved performance compared to the rest of BL-U approaches while being completely blind to the protected attribute labels. Moreover, for the *Heavy-Makeup* task, FLAC-B notably outperforms the BL-A methods across both *unbiased* and *bias-conflict* samples.

Table IV presents the results of FLAC compared to other methods on the UTKFace dataset with *race* and *age* as protected attributes. For both protected attributes, the proposed method surpasses the best performing competing methods on both the *unbiased* (i.e., +1% and +1.6%) and *bias-conflict* test sets (i.e., 4% and 9.4%), while it outperforms the state-of-the-art BL-U methods by 5%-6.1% and 13%-21.9% on the *unbiased* and *bias-conflict* test sets, respectively. Here, it is noteworthy that FLAC exhibits very high accuracy (i.e., 81.1%) on the most challenging test set (i.e., *bias-conflict* test set with *age* as the protected attribute), whereas the best BL-A method achieves 71.7%. In addition, FLAC-B not only maintains high unbiased accuracy but also achieves notable improvements in the performance on bias-conflicting samples, even though it employs the Vanilla model as a bias-capturing model.

Furthermore, Fig. 4 visualizes the 8 most similar samples to a query image that represents a minority group for Vanilla and FLAC. In this example, we employed the biased UTKFace dataset with *gender* as target and *race* as protected attribute, where the minority group is the non-white males, due to the high correlation between the males and the white race in the training data. By inspecting Fig. 4(a) and (b), we may observe that given a query depicting a *non-white male*, Vanilla's similar representations correspond mostly to people of color (both males and females) while retrieved images based on the FLAC's representations depict males of various races, i.e., they do not encode racial information. The visualization provided in Fig. 4 indicates the effectiveness of FLAC in (i) disassociating the representations from the *race* attribute and (ii) bringing the representations with the same *gender* closer to each other.

The last evaluation scenario does[6] not refer to societal biases, but biases that are introduced from the image's background or texture. The results on the Corrupted-Cifar10 for 4 different correlation ratios are presented in Table V. It is noteworthy that FLAC consistently outperforms the second best performing method for all the different correlation ratios, namely 2.9%, 4.2%, 2.8%, and 4.1% for q equal to 0.95, 0.98, 0.99, and 0.995, respectively. Table VI presents the results of the experiments conducted on the 9-Class ImageNet dataset. In this scenario, the

[6]Reimplemented based on the code provided by the authors.

TABLE V
EVALUATION ON CORRUPTED-CIFAR10

methods	L_{task}	BL-U	q			
			0.95	0.98	0.99	0.995
Vanilla	CE	✓	39.4±0.6	30.1±0.7	25.8±0.3	23.1±1.2
EnD	CE	✗	36.6±4.0	34.1±4.8	23.1±1.1	19.4±1.4
FairKL	Con.	✗	50.7±0.9	41.5±0.4	36.5±0.4	33.3±0.4
HEX	CE	✓	16.0±0.6	15.2±0.5	14.8±0.4	13.9±0.0
ReBias	CE	✓	43.4±0.4	31.7±0.4	25.7±0.2	22.3±0.4
LfF	CE	✓	50.3±1.6	39.9±0.3	33.1±0.8	28.6±1.3
DisEnt	CE	✓	51.1±1.3	41.8±2.3	36.5±1.8	30.0±0.7
FLAC-B	Con.	✓	**53.0**±0.7	**46.0**±0.2	**39.3**±0.4	**34.1**±0.5

Underlined and dotted-underlined values refer to the second best performing BL-U and the best performing BL-A methods, respectively. Con. Refer to contrastive losses.

TABLE VI
EVALUATION OF THE PROPOSED METHOD ON THE BIASED AND THE UNBIASED 9-CLASS IMAGENET TEST SETS AND THE IMAGENET-A

methods	$\mathcal{L}_{task}$	biased	unbiased	ImageNet-A
Vanilla	CE	94.0±0.1	92.7±0.2	30.5±0.5
LM	CE	79.2±1.1	76.6±1.2	19.0±1.2
Rubi	CE	93.9±0.2	92.5±0.2	31.0±0.2
ReBias	CE	94.0±0.2	92.7±0.2	30.5±0.2
LfF	CE	91.2±0.1	89.6±0.3	29.4±0.8
SoftCon	Con.	95.3±0.2	94.1±0.3	34.1±0.6
FairKL-lu	Con.	95.1±0.1	94.8±0.3	35.7±0.5
FLAC-B	CE	95.5±0.2	**95.2**±0.2	37.6±0.6
FLAC-B	Con.	**95.7**±0.2	**95.2**±0.2	**37.9**±0.9

FairKL-lu refers to the BL-U version of FairKL.

bias is not categorical, thus only BL-U methods can be applied. As presented in Table VI, FLAC outperforms the state-of-the-art on all three test sets. As in previous experiments, FLAC demonstrates 2.2% improvements in terms of accuracy on the most challenging test set (i.e., ImageNet-A).

VI. ABLATION STUDY

FLAC is designed to be applied only on the meaningful pairs of samples (see Section III-D), which is not the case for other BL-U approaches. Particularly, other works [16], [17] involve all the possible sample pairs by assigning weights based on their similarities. However, the meaningful pairs of samples are only a very small subset of the set of all the possible pairs, especially in extremely biased data, such as the Biased-MNIST with $q = 0.999$ or the CelebA with *HeavyMakeup* as target. As presented in Fig. 5, the maximum number of sample pairs in the Biased-MNIST with $q = 0.99$ that satisfy (1) is 366 (i.e., only 0.022% of all pairs), while the number of all the possible sample pairs is 128^2. Involving all sample pairs in the bias mitigation procedure makes competing BL-U methods much more sensitive to scenarios where only few sample pairs in the dataset are responsible for the emergence of bias. Table VII compares the performance of FLAC and two state-of-the-art BL-U approaches, namely SoftCon [16] and FairKL-lu [17], w/ and w/o leveraging the sampling based on (1). It becomes obvious that using (1) can significantly improve the performance of other BL-U methods, while FLAC w/ (1) still exhibits the best performance. Furthermore, the importance of involving both terms of (1) as analyzed in Section III-D is experimentally

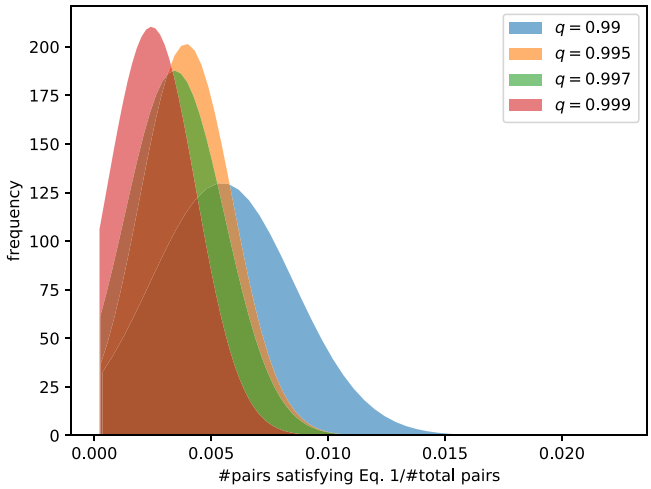


Fig. 5. The frequency of the number of sample pairs belonging to $\mathcal{S}$ per batch ($N = 128$) for Biased-MNIST training set with $p \in \{0.99, 0.995, 0.997, 0.999\}$.

TABLE VII
PERFORMANCE OF STATE-OF-THE-ART BL-U METHODS W/O OR W/ ADOPTING (1)

method	$\mathcal{L}_{task}$	accuracy w/o Eq. (1)	accuracy w/ Eq. (1)
SoftCon	Con.	65.0	84.0
FairKL-lu	Con.	13.7	73.9
FLAC	CE	24.1	89.3
FLAC	Con.	28.2	**94.1**

Results pertain to BiasedMNIST with $q = 0.999$

TABLE VIII
THE IMPACT OF THE TWO TERMS OF (1) ON THE PERFORMANCE OF FLAC

Equation (1) variants	acc. (CE)	acc. (Con.)
$y_i \neq y_j \wedge t_i = t_j$	10.1	29.1
$y_i = y_j \wedge t_i \neq t_j$	10.5	20.6
$(y_i = y_j \wedge t_i \neq t_j) \vee (y_i \neq y_j \wedge t_i = t_j)$	**89.3**	**94.1**

Results pertain to BiasedMNIST with $q = 0.999$

TABLE IX
THE IMPACT OF KERNEL FUNCTION ON THE PERFORMANCE OF FLAC

kernel	accuracy (CE)	accuracy (Con.)
cosine	60.4	82.8
RBF	83.4	90.7
student's t	**89.3**	**94.1**

Results pertain to BiasedMNIST with $q = 0.999$

confirmed in Table VIII, which presents the performance of FLAC using only the first (i.e., pairs with the same target and different protected attribute label) or the second term (i.e., pairs with different target and same protected attribute label) of (1).

Table IX presents the impact of several kernel functions on the performance of the proposed method. The kernel function should be carefully selected as it can significantly affect the model's performance. For instance, although the cosine kernel fits well in retrieval tasks, it is not effective enough on classification tasks as previous studies have noticed [42], which is also reflected in Table IX. Furthermore, Radial Basis Function (RBF) kernels

TABLE X
PERFORMANCE OF FLAC WITH MSE, KULLBACK-LEIBLER, AND JEFFREYS DIVERGENCE AS LOSS FUNCTIONS

loss	accuracy (CE)	accuracy (Con.)
MSE	12.1	85.6
Kullback-Leibler divergence	85.2	89.2
Jeffreys divergence (Eq. (6))	**89.3**	**94.1**

Results pertain to BiasedMNIST with $q = 0.999$

TABLE XI
THE IMPACT OF FLAC WHEN THE TRAINING DATA IS UNBIASED

method	accuracy (CE)	accuracy (Con.)
Vanilla	99.3	-
FLAC@α=1	99.4	99.3
FLAC@α=100	99.1	99.1
FLAC@α=1000	98.2	98.9

Results for BiasedMNIST with $q = 0.1$

constitute a common kernel option, but tuning them is often difficult. Finally, the student's t kernel, which is considered as a good option for classification tasks, is shown to achieve the best performance compared to other kernels in the conducted experiments.

Table X presents the evaluation of FLAC using different loss functions, namely the MSE, Kullback-Leibler divergence, and Jeffreys divergence. As expected, using a common regression loss, such as MSE, is not a good choice to mitigate bias, while Kullback-Leibler and Jeffreys divergences lead to highly accurate models, with Jeffreys divergence leading to the best results.

In order to investigate whether the proposed method can have a negative impact on the model's performance when the training data is unbiased, we further evaluate the proposed method on a dataset that does not suffer from any attribute-label associations (i.e., Biased-MNIST with $q = 0.1$). Table XI presents the performance of FLAC for $\alpha \in \{1, 100, 1000\}$: for $\alpha = 1$ and $\alpha = 100$ FLAC does not affect the model's performance, while for the extreme value $\alpha = 1000$ only a small drop in accuracy is noticed.

VII. CONCLUSION

In this paper, we introduce FLAC, a bias mitigation approach that leverages the representations of a bias-capturing classifier for enabling the main model to learn fair representations without being aware of the protected attribute labels. In particular, FLAC aims to minimize the MI between the main model's representations and the protected attribute without taking advantage of the protected attribute labels. To this end, FLAC only leverages the under-represented samples that can effectively contribute to minimizing MI. By doing so, the derived fair representations do not capture any information related to the protected attributes. The proposed approach demonstrates superior performance compared to state-of-the-art in a wide range of experiments on both datasets with artificially injected bias and established computer vision datasets. As regards the limitations of the proposed approach, it should be highlighted that FLAC necessitates the presence of some bias-conflicting samples in the training data and thus cannot be applied in 100% correlation scenarios. Furthermore, its effectiveness diminishes for batch sizes equal to or lower than 32, and it does not account for multiple sources of bias. Thus, as a future work, we consider enabling FLAC to be applied in a multi-attribute fairness scenario. Furthermore, taking into account that bias is uniformly distributed across classes in the existing fairness benchmarks, another future work could be the exploration of fairness-aware approaches performance on datasets with non-uniform bias distribution. Finally, using FLAC for tasks beyond classification (e.g., retrieval), is a potential subject for future work.

REFERENCES

[1] S. Barocas, M. Hardt, and A. Narayanan, Fairness and machine learning: Limitations and opportunities. fairmlbook.org, 2019, [Online]. Available: http://www.fairmlbook.org

[2] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.

[3] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, 2013.

[5] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10781–10790.

[6] S. Fabbrizzi, S. Papadopoulos, E. Ntoutsi, and I. Kompatsiaris, "A survey on bias in visual datasets," *Comput. Vis. Image Understanding*, vol. 223, 2022, Art. no. 103552.

[7] M. Frisella, P. Khorrami, J. Matterer, K. Kratkiewicz, and P. Torres-Carrasquillo, "Quantifying bias in a face verification system," in *Computer Sciences & Mathematics Forum*. Basel, Switzerland: MDPI, 2022, Art. no. 6.

[8] S. Glüge, M. Amirian, D. Flumini, and T. Stadelmann, "How (not) to measure bias in face recognition networks," in *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*. Berlin, Germany: Springer, 2020, pp. 125–137.

[9] J. G. Cavazos, P. J. Phillips, C. D. Castillo, and A. J. O'Toole, "Accuracy comparison across face recognition algorithms: Where are we on measuring race bias?," *IEEE Trans. Biometrics Behav. Identity Sci.*, vol. 3, no. 1, pp. 101–111, Jan. 2021.

[10] T. Sixta, J. Jacques Junior, P. Buch-Cardona, E. Vazquez, and S. Escalera, "Fairface challenge at ECCV 2020: Analyzing bias in face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 463–481.

[11] T. Xu, J. White, S. Kalkan, and H. Gunes, "Investigating bias and fairness in facial expression recognition," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2020, pp. 506–523.

[12] X. Zheng, Y. Guo, H. Huang, Y. Li, and R. He, "A survey of deep facial attribute analysis," *Int. J. Comput. Vis.*, vol. 128, pp. 2002–2034, 2020.

[13] H. Zhao and G. J. Gordon, "Inherent tradeoffs in learning fair representations," *J. Mach. Learn. Res.*, vol. 23, pp. 1–26, 2022.

[14] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim, "Learning not to learn: Training deep neural networks with biased data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9012–9020.

[15] E. Tartaglione, C. A. Barbano, and M. Grangetto, "End: Entangling and disentangling deep representations for bias correction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13508–13517.

[16] Y. Hong and E. Yang, "Unbiased classification through bias-contrastive and bias-balanced learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 26449–26461.

[17] C. A. Barbano, B. Dufumier, E. Tartaglione, M. Grangetto, and P. Gori, "Unbiased supervised contrastive learning," 2022, *arXiv:2211.05568*.

[18] C. Clark, M. Yatskar, and L. Zettlemoyer, "Don't take the easy way out: Ensemble based methods for avoiding known dataset biases," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4069–4082.

[19] R. Cadene et al., "RUBi: Reducing unimodal biases for visual question answering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 841–852.

[20] H. Bahng, S. Chun, S. Yun, J. Choo, and S. J. Oh, "Learning de-biased representations with biased representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 528–539.
[21] J. Nam, H. Cha, S. Ahn, J. Lee, and J. Shin, "Learning from failure: Debiasing classifier from biased classifier," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 20673–20684.
[22] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 3730–3738.
[23] Z. Zhifei, S. Yang, and Q. Hairong, "Age progression/regression by conditional adversarial autoencoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4352–4360.
[24] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and surface variations," 2018, *arXiv: 1807.01697*.
[25] S. Yan, D. Huang, and M. Soleymani, "Mitigating biases in multimodal personality assessment," in *Proc. Int. Conf. Multimodal Interaction*, 2020, pp. 361–369.
[26] A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto, "Robot learning in homes: Improving generalization and reducing dataset bias," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9112–9122.
[27] D. Liu, X. Qu, and W. Hu, "Reducing the vision and language bias for temporal sentence grounding," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 4092–4101.
[28] P. Sattigeri, S. C. Hoffman, V. Chenthamarakshan, and K. R. Varshney, "Fairness GAN," *Stat*, vol. 1050, 2018, Art. no. 24.
[29] D. Xu, S. Yuan, L. Zhang, and X. Wu, "FairGAN: Fairness-aware generative adversarial networks," in *Proc. 2018 IEEE Int. Conf. Big Data*, 2018, pp. 570–575.
[30] E. Ntoutsi et al., "Bias in data-driven artificial intelligence systems—an introductory survey," *Wiley Interdiscipl. Reviews: Data Mining Knowl. Discov.*, vol. 10, no. 3, 2020, Art. no. e1356.
[31] Z. Wang et al., "Towards fairness in visual recognition: Effective strategies for bias mitigation," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2020, pp. 8919–8928.
[32] M. Alvi, A. Zisserman, and C. Nellåker, "Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 556–572.
[33] Q. Xie, Z. Dai, Y. Du, E. Hovy, and G. Neubig, "Controllable invariance through adversarial feature learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 585–596.
[34] T. Wang, J. Zhao, M. Yatskar, K.-W. Chang, and V. Ordonez, "Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5310–5319.
[35] J. Song, P. Kalluri, A. Grover, S. Zhao, and S. Ermon, "Learning controllable fair representations," in *Proc. 22nd Int. Conf. Artif. Intell. Statisti.*, 2019, pp. 2164–2173.
[36] T. Adel, I. Valera, Z. Ghahramani, and A. Weller, "One-network adversarial fairness," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 2412–2420.
[37] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with hilbert-schmidt norms," in *Proc. 16th Int. Conf. Algorithmic Learn. Theory*, 2005, pp. 63–77.
[38] J. Nam, J. Kim, J. Lee, and J. Shin, "Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation," in *Proc. Int. Conf. Learn. Representations*, 2022.
[39] S. Jung, S. Chun, and T. Moon, "Learning fair classifiers with partially annotated group labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10348–10357.
[40] J. Lee, E. Kim, J. Lee, J. Lee, and J. Choo, "Learning debiased representation via disentangled feature augmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 25123–25133.
[41] J. Lee, J. Park, D. Kim, J. Lee, E. Choi, and J. Choo, "Revisiting the importance of amplifying bias for debiasing," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 14974–14981.
[42] N. Passalis, M. Tzelepi, and A. Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2339–2348.
[43] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 268–284.
[44] I. Sarridis, C. Koutlis, S. Papadopoulos, and I. Kompatsiaris, "Indistill: Transferring knowledge from pruned intermediate layers," 2022, *arXiv:2205.10003*.
[45] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Comput. Appl.*, vol. 24, pp. 175–186, 2014.
[46] A. Kraskov, H. Stögbauer, R. G. Andrzejak, and P. Grassberger, "Hierarchical clustering using mutual information," *Europhysics Lett.*, vol. 70, no. 2, 2005, Art. no. 278.
[47] W. Li, Z. Liang, J. Neuman, J. Chen, and X. Cui, "Multi-generator GAN learning disconnected manifolds with mutual information," *Knowl.-Based Syst.*, vol. 212, 2021, Art. no. 106513.
[48] K. Lum and J. Johndrow, "A statistical framework for fair predictive algorithms," 2016, *arXiv:1610.08077*.
[49] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *J. Mach. Learn. Res.*, vol. 3, no. Mar, pp. 1415–1438, 2003.
[50] S. M. Ali and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another," *J. Roy. Statist. Society: Ser. B. Methodological*, vol. 28, no. 1, pp. 131–142, 1966.
[51] M. Boudiaf et al., "A unifying mutual information view of metric learning: Cross-entropy vs. pairwise losses," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., 2020, pp. 548–564.
[52] R. Das and S. Chaudhuri, "On the separability of classes with the cross-entropy loss function," 2019, *arXiv: 1909.06930*.
[53] Y. LeCun, "The MNIST database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. 2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[55] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15257–15266.
[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[57] W. Brendel and M. Bethge, "Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet," 2019, *arXiv: 1904.00760*.
[58] P. Khosla et al., "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 18661–18673.
[59] P. Meier, J. Sacks, and S. L. Zabell, "What happened in hazelwood: Statistics, employment discrimination, and the 80% rule," *Amer. Bar Found. Res. J.*, vol. 9, no. 1, pp. 139–186, 1984.
[60] P. Janssen and B. M. Sadowski, "Bias in algorithms: On the trade-off between accuracy and fairness," in *Proc. 23rd Biennial Conf. Int. Telecommun. Soc.*, Gothenburg, Sweden, Jun. 2021, pp. 21–23.
[61] E. Krasanakis, E. Spyromitros-Xioufis, S. Papadopoulos, and Y. Kompatsiaris, "Adaptive sensitive reweighting to mitigate bias in fairness-aware classification," in *Proc. World Wide Web Conf.*, 2018, pp. 853–862.

Ioannis Sarridis received the bachelor's and master's degrees from the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2018 and 2020, respectively. He is currently working toward the PhD degree with the Harokopio University of Athens and is a research associate with the Centre for Research and Technology Hellas, Thessaloniki, Greece. His research interests include AI fairness, computer vision, deep learning, and hypergraph learning.

Christos Koutlis received the bachelor's degree in mathematics and master's degree in statistics and modeling from the Department of Mathematics, Aristotle University of Thessaloniki, in 2012 and 2014, respectively, and the PhD degree in multi-variate time-series analysis from the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, in 2017. He is a postdoctoral researcher with the Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece. He has co-authored more than 20 papers published in international scientific journals and conferences. His recent research interests include deep learning based visual and multimodal media analysis as well as multivariate time series trend analysis.

Symeon Papadopoulos received the PhD degree in computer science from the Aristotle University of Thessaloniki in 2012 on the topic of Knowledge discovery from large-scale mining of social media content. He is a principal researcher with the Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece. He has co-authored more than 40 papers in refereed journals, 10 book chapters and 130 papers in international conferences, 3 patents, and has edited two books. He has participated in and coordinates a number of relevant EC FP7, H2020 and Horizon Europe projects in the areas of media convergence, social media and artificial intelligence. He is leading the Media Analysis, Verification and Retrieval (MeVer) Group, and is a co-founder of the Infalia Private Company, a spin-out of CERTH-ITI.

Christos Diou (Member, IEEE) received the diploma degree in electrical and computer engineering and the PhD degree in analysis of multimedia with machine learning from the Aristotle University of Thessaloniki. He is an assistant professor in artificial intelligence and machine learning with the Department of Informatics and Telematics, Harokopio University of Athens. He has co-authored more than 80 publications in international scientific journals and conferences and is the co-inventor in one patent. His recent research interests include robust machine learning algorithms that generalize well, the interpretability of machine learning models, as well as the development of machine learning models for the estimation of causal effects from observational data. He has more than 15 years of experience participating and leading European and national research projects, focusing on applications of artificial intelligence in healthcare.