

AperTO - Archivio Istituzionale Open Access dell'Università di Torino

Mining bias-target Alignment from Voronoi Cells

This is the author's manuscript

Original Citation:

Availability:

This version is available <http://hdl.handle.net/2318/1992971> since 2024-07-04T12:19:01Z

Publisher:

IEEE COMPUTER SOC

Published version:

DOI:10.1109/iccv51070.2023.00456

Terms of use:

Open Access

Anyone can freely access the full text of works made available as "Open Access". Works made available under a Creative Commons license can be used according to the terms and conditions of said license. Use of all other works requires consent of the right holder (author or publisher) if not exempted from copyright protection by the applicable law.

(Article begins on next page)

Mining bias-target Alignment from Voronoi Cells

Rémi Nahon

Van-Tam Nguyen

Enzo Tartaglione

LTCI, Télécom Paris, Institut Polytechnique de Paris

remi.nahon@telecom-paris.fr

Abstract

Despite significant research efforts, deep neural networks are still vulnerable to biases: this raises concerns about their fairness and limits their generalization. In this paper, we propose a bias-agnostic approach to mitigate the impact of bias in deep neural networks. Unlike traditional debiasing approaches, we rely on a metric to quantify “bias alignment/misalignment” on target classes, and use this information to discourage the propagation of bias-target alignment information through the network. We conduct experiments on several commonly used datasets for debiasing and compare our method to supervised and bias-specific approaches. Our results indicate that the proposed method achieves comparable performance to state-of-the-art supervised approaches, although it is bias-agnostic, even in presence of multiple biases in the same sample.

1 Introduction

Deep Neural Networks (DNNs) are known today for their high performance and resilience in many areas of computer vision, such as image classification, semantic segmentation, and object detection, used in areas ranging from self-driving vehicles to face recognition or surgical guidance. However, it is well known that their tendency to rely heavily on any type of correlation present in the training data exposes them to potential pitfalls [17, 2, 39]: some “spurious correlations” may be mistakenly learned by the DNN. These can take over the role of *biases* [35].

Learned biases may decrease the generalization of the DNN [17, 2, 25, 30, 4, 9]. For example, if a DNN has learned to distinguish airplanes flying in the sky from boats sailing in the ocean, the model will likely use the background as a base for its classification: detecting it instead of learning the vehicle shape is a much simpler task. However, the model does not generalize to scenarios such as a landing seaplane. Differently from domain adaptation [10, 29, 36], where the objective is to learn general features compensating the domain shift, or to adapt the extracted features to different domains, the goal of debiasing is to discourage the learning of spurious correlations.

Many current debiasing approaches rely on prior information about the bias, such as the existence of an auxiliary label indicating some side information, the presence of bias(es) or their quality [34, 5, 4, 9, 37, 12]. However, obtaining these labels or information about the nature of the bias can be either very expensive (due to annotation costs) or very noisy: this is what motivates the development of bias-agnostic approaches. Recent works have shown that bias features are learned “early” [30, 26]: there are bias-target *aligned* samples, for which the bias is learned, and the performance on the train set increases, and some *disaligned* ones, for which the prediction is wrong. Since bias-agnostic approaches delve into the biased information from the training set, it is common to amplify the first features learned using Generalized Cross-Entropy (GCE) [40] and then discourage their learning in an “unbiased” model. However, there is no guarantee that the very first features learned are the biased ones: detecting them agnostically and effectively remains an open question.

In this work, we propose a method that identifies the best time to extract bias-target alignment information by observing the relative distance of misclassified samples to the nearest Voronoi hyperplane of the correct target class. We use this information to train an unbiased model, from which we give higher weight to bias-misaligned samples, and remove the bias-alignment information from the bottleneck layer (Fig. 1).

At a glance, our contributions are the following:

- we propose a bias-agnostic approach which indicates, during the training of a vanilla model, when to extract bias-target alignment information: more precisely, we observe the distance of misclassified samples to the closest Voronoi hyperplane of the correct target class (Sec. 3.3);
- we use the bias misalignment information to weight the loss contribution of every single sample: this will favor the learning of misclassified samples in the vanilla setup (Sec. 3.4.1);

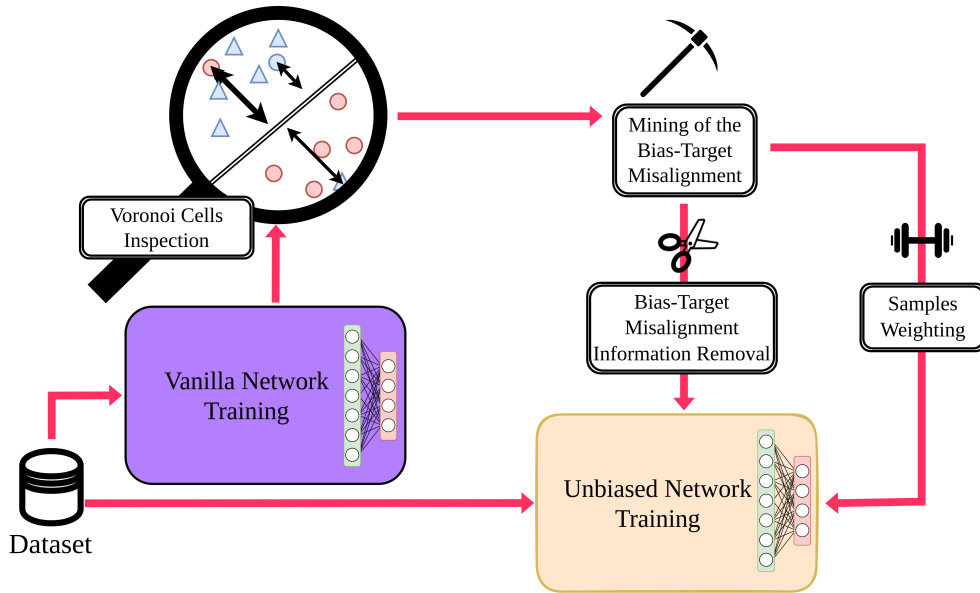


Figure 1: Our proposed approach to agnostically remove the bias.

- we also propose an approach to eliminate bias misalignment information: specifically, we minimize the bias alignment information which is extractable at the bottleneck of the DNN, conditioned to the bias target alignment (Sec. 3.4.2);
- we study the behavior on several datasets typically used for debiasing and compare both supervised and bias-agnostic approaches: although our proposed technique is bias-agnostic, its performance is comparable to supervised approaches (Sec. 4.3).

2 Related works

Avoiding algorithmic bias plays an important role in artificial intelligence (AI) ethics, and the area of research which focuses on this is *fairness*. DNN predictions are “unfair” if they are based on specific sets of features and classes that will unfairly impact certain groups, according to some ethical principles [13, 15, 7, 23]. Some fairness metrics measure this type of inequality: Demographic Parity, Equalized Odds, or Equal Opportunity are some representative examples [18, 13, 16]. Fairness and debiasing are intrinsically related, but while the former explicitly highlights a measure of fairness, the latter maximizes the performance on an “unbiased” dataset, without necessarily declaring a fairness metric. Below, we will review debiasing approaches, which can be divided into *supervised* methods (where we have access to a “bias label”), and *unsupervised* methods.

2.1 Supervised methods

Supervised debiasing methods are divided into three categories: pre-processing methods, which modify the dataset prior to classification; in-processing methods, which modify the learning process of the model; and post-processing methods, which directly modify the output of the DNN.

Preprocessing methods. Among the most used preprocessing methods in the literature, driven data augmentation plays a prominent role. Generative Adversarial Networks (GANs) are widely used to generate realistic images: StyleGANs [11] is indeed one of the mostly used GANs in this context. For example, Kang *et al.* [24] used it to generate handwritten text in specific styles. In image classification, Geirhos *et al.* [17] used style transfer to augment ImageNet with texture-bias-conflicting elements to create a more texture-balanced dataset.

Postprocessing methods. These methods have the advantage of neither re-training models nor requiring additional data for the training. With their Reject Option Classification, for example, Kamiran *et al.* [22] proposed to take the samples classified with the most uncertainty (outside a predefined confidence margin) and to change their class to a lower Disparate Impact. In this same context, Equalized Odds Postprocessing proposed by Hardt *et al.* [18] maximizes

the Equalized Odds metric. Despite the potential advantages of these approaches, a major drawback lies in the low degrees of freedom for the corrections (since they can only access post-classification information), which limits their practical effectiveness.

In-processing: debiasing within training. Most of the debiasing methods in the literature work directly on the model, learning from a biased dataset. In general, unbiased elements are weighted more than biased elements. This simple yet effective approach is nowadays very popular in supervised setups [21]. Other methods tackle supervised debiasing by adding regularization terms during the training of the deep model, which is the case of methods such as EnD [34] and FairKL [5]. Another intuitive approach relies upon simply removing the biased features from each sample in the dataset and performing the so-called *fairness by blindness*. However, the phenomenon known as *encoding redundancy* [18] states that information is very rarely encoded only once in the data [31], so removing a single value or label is probably not sufficient to remove the effect of the bias on classification.

2.2 Unsupervised methods

Some recent methods do not rely on bias labels because they can be difficult to obtain on real-life datasets and we will refer to them as “unsupervised” or “bias-agnostic”. All of these approaches follow a general scheme, which is typically divided into two phases: *bias inference*, where a first model, often called “bias capturing”, aims to capture biases in the data; and *bias mitigation*, where a second model is trained to avoid the biases captured by the first model. These approaches rely on prior knowledge, which may be more or less specific to the target task.

Bias is in the texture. Some works focus on the bias specifically present in texture, as it is prominent in image classification [17]. Rebias [4] promoted learning with representations that are maximally different from using small receptive fields in convolutional layers. These are biased-by-design, toward learning specific textures. For the same texture debiasing task, HEX [37] proposed to use the gray-level co-occurrence matrix and to promote representation independent of colors.

Bias generates imbalances between groups. Some unsupervised approaches involve finding the bias groups that optimize some fairness metric and train the model to have representations orthogonal to those inferred by the biases. With DebiAN [27], Li *et al.* for example proposes a method that alternates between the training of bias-capturing and unbiased models, minimizing the Equal Opportunity fairness metric. In EIL [14], Creager *et al.* identified biases by finding the groups that maximize violation of an invariance principle measured by the objective function IRMv1 [3]. Similarly, PGI [1] built upon EIL by minimizing the KL-Divergence of the prediction over these groups.

Bias is learned early. Some recent methods are based on the assumption that bias features are easy to learn. These features can be extracted at a given point, at the beginning of the training. With LfF [30], Nam *et al.* proposed a loss reweighing method based on this assumption: they train a biased neural network and amplify its early stages prediction. In parallel, they train a debiased model weighting “difficult samples”. Based on this, with DFA [26], Lee *et al.* performed data augmentation attempting to disentangle bias features from intrinsic features through latent representations of the bias-capturing and unbiased models.

The closest competing strategy to the one we propose is LfF. Differently from [30], our main assumption is not that the first features learned by the model are biased: we assume that the model, at some point, will adapt to the bias and that it is possible to identify this moment by examining the latent representation of the dataset. This particular point can occur at any time during the training, and so emphasizing the earlier choices of the model can prevent it from efficiently adapting to the bias. Moreover, unlike [30], we do not seek to extract the information from the bias, but its alignment with target classes, which allows our approach to easily scale to multi-biased setups.

3 Proposed Method

3.1 Overview of the proposed method

Let us consider a supervised learning setup, where we have a dataset \mathcal{D} containing n input samples $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathcal{X}$, each associated to a ground truth target label $(\hat{y}_1, \dots, \hat{y}_n) \in \mathcal{Y}$. A given deep neural network \mathcal{M} , trained for e epochs, produces an output $y_{e,i}$ given some \mathbf{x}_i , and is typically trained to match $\hat{y}_i \forall i$ through the minimization of a loss function $\mathcal{L}(y_{e,i}, \hat{y}_i)$. Unfortunately, this learning process does not impose any prior on the specific subset of features that are extracted: these can lead the prediction over unseen data to be biased: we want to fight this effect.

Fig. 2 provides an overview of the proposed debiasing approach. First, the bias is inferred by the learning of a vanilla model: at the end of each epoch (or after a few iterations), the target class centroids and the decision hyperplanes are computed from the well-classified samples at the bottleneck layer (Sec. 3.2), and the distance of the misclassified

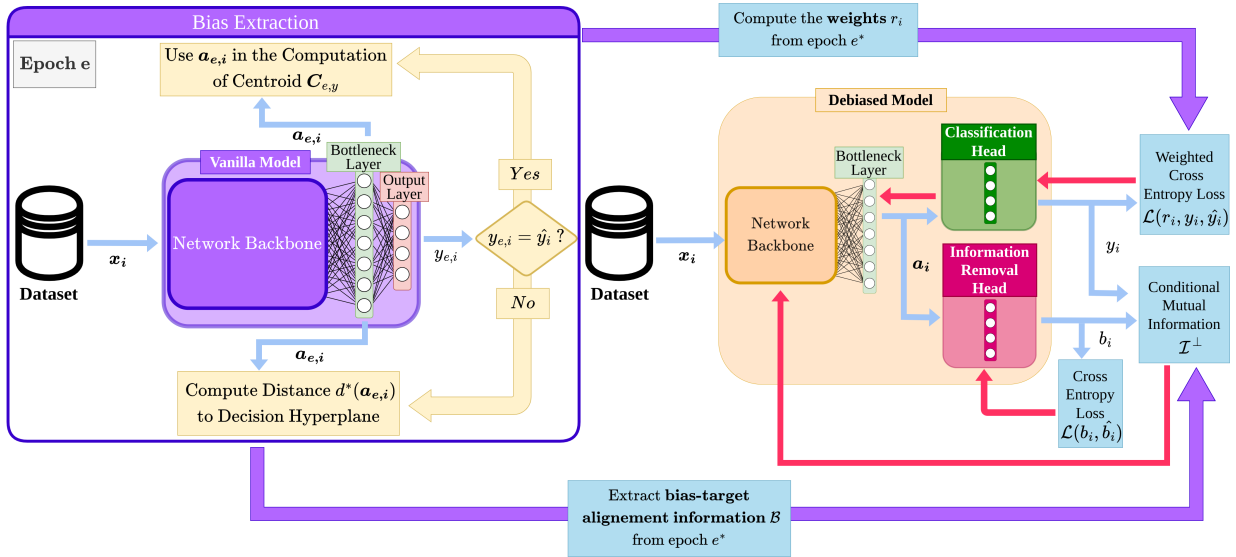


Figure 2: Overview of the proposed debiasing approach: the bias is first extracted (left) and then the unbiased model is trained (right). Blue arrows represent forward propagation while red arrows represent backpropagation.

samples to the Voronoi cell of the correct class is computed to find the epoch e^* when the bias-target alignment is maximally learned (Sec. 3.3). Then, a debiasing process follows (Sec. 3.4): from the distances gathered from the previous step, we assign each sample a weight, which will be used in the weighted cross-entropy loss (Sec. 3.4.1). In addition, at the bottleneck layer, we also minimize the information about bias misalignments: this favors the unbiasedness of the classification head (Sec. 3.4.2). In the rest of this section, we will detail all the steps of our proposed technique.

3.2 Bottleneck latent representation

The debiasing method we propose stems from the concept of latent representation: the output of each layer of a DNN consists of a representation of the input x_i . Thus, the classification phase, which takes place just before the output of the model, consists in partitioning the feature space into each of the different classes. Therefore, the output of the *bottleneck layer* (the output of the backbone) is the compressed representation of the input sample, which is often referred to as *latent representation*. Therefore, each x_i has a vector of latent attributes $\mathbf{a}_{e,i} = (a_{e,i,1}, \dots, a_{e,i,K}) \in \mathbb{R}^K$ associated to a specific epoch (or iteration) e for the model \mathcal{M} : this forms its latent bottleneck representation.

We define $\mathcal{D}_e^{\parallel}$ the set of samples well classified by the model \mathcal{M} at epoch e , and \mathcal{D}_e^{\perp} the misclassified samples. For each t -th target class, it is possible to define a *class centroid* $\mathbf{C}_{e,t}$ as the average of the bottleneck representations of each well-classified samples of the t -th target class:

$$\mathbf{C}_{e,t} = \frac{1}{\|\mathcal{D}_{e,t}^{\parallel}\|_0} \sum_{i \in \mathcal{D}_{e,t}^{\parallel}} \mathbf{a}_{e,i}, \quad (1)$$

where $\mathcal{D}_{e,t}^{\parallel}$ is the subset of correctly classified samples for the t -th class, and $\|\mathcal{D}\|_0$ counts the elements in \mathcal{D} . Such centroids are proxies for the representations of the correctly classified elements of the class by the model. Let us define *decision hyperplane* $\mathbf{H}_{e,i,j}$ as the hyperplane equidistant from $\mathbf{C}_{e,i}$ and $\mathbf{C}_{e,j}$ in the bottleneck representation space of \mathcal{M} : $\mathbf{H}_{e,i,j}$ is a proxy of the Voronoi decision boundary.

In Fig. 3, we have two target classes (the triangles and the squares) and the bias is pictured as the color (blue and red). The first image on the left shows the latent representation of the dataset by the model at its initialization: the samples are scattered randomly in the feature space, a first classification occurs, materialized by the decision hyperplane $\mathbf{H}_{0,\Delta,\circ}$. Ideally, as shown in the second picture for Fig. 3, a deep learning model should minimize the

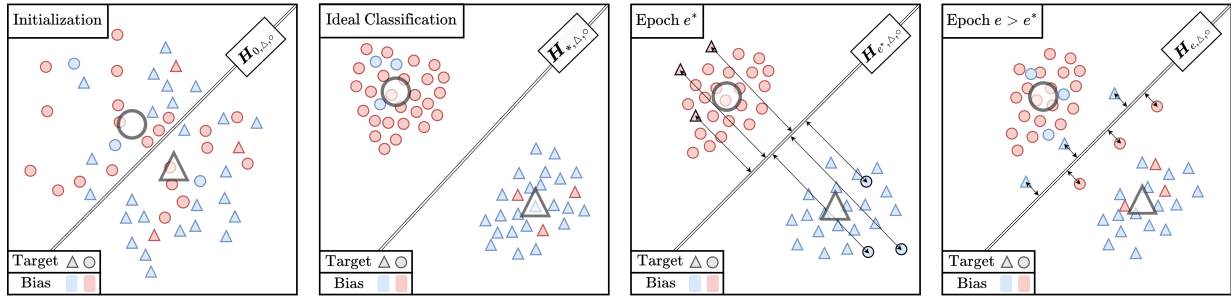


Figure 3: Representation of latent representations of a dataset at different learning stages. The target classes are shapes and the bias is color. Arrows represent the distance between specific elements and the Voronoi hyperplane.

intra-class distance and maximize the inter-class one. However, in the presence of bias, another kind of attractor can emerge: the bias-conflicting elements will be attracted by the wrong class. The more the model is biased, the more the sample clusters that form will represent the biased classes more than the target class. For instance, in the third picture, the samples are clustered by color and not by shape: the red triangles have been attracted by the circle class that is correlated with the color red. If the model is, then, over-parametrized, the biased elements will be attracted as well towards the right centroid (fourth figure): although a set of bias-misaligned features are learned, the model still holds biased ones, which leads to a subpar generalization performance: our first goal will be, hence, to detect the e^* moment of the learning where it is possible to extract the bias-target misalignment information.

3.3 Bias alignment capture

To distinguish between bias-misaligned samples and bias-aligned ones, we assume that, after a few learning steps, the farther a misclassified sample is distant from its target Voronoi cell, the more it has been strongly pulled by an attractor. Such an attractor, since it is not its target class centroid, can be considered as resulting from some bias. Hence, when the average distance between the misclassified samples and their target Voronoi cell reaches its maximum, the model has learned bias features. To select the exact moment when to extract the bias-target alignment, we are looking for the epoch

$$e^* = \operatorname{argmax}_e \frac{\sum_i d^*(\mathbf{a}_{e,i})}{\|\mathcal{D}_e^\perp\|_0} \quad (2)$$

where

$$d^*(\mathbf{a}_{e,i}) = \begin{cases} 0 & \text{if } y_{e,i} = \hat{y}_i \\ \|\mathbf{a}_{e,i} - \mathbf{H}_{e, \mathcal{C}_{e, y_{e,i}}, \mathcal{C}_{e, \hat{y}_i}}\|_2 & \text{if } y_{e,i} \neq \hat{y}_i, \end{cases} \quad (3)$$

where $\|\cdot\|_2$ indicates the ℓ_2 norm. This scenario is visualized in Fig. 3 (bottom-left). At this point, we can collect bias-target alignment information $(y_{e^*,1} = \hat{b}_1, \dots, y_{e^*,n} = \hat{b}_n) \in \mathcal{B}$. After this step, we can hereby identify the subset of bias-target misaligned samples \mathcal{D}^\perp , and the set of bias-target aligned ones \mathcal{D}^\parallel . Given that vanilla learning strategies employ weight-decay, for which the distances tend to diminish when reaching the loss minimum, we propose to modify (3) as

$$d^*(\mathbf{a}_{e,i}) = \begin{cases} 0 & \text{if } y_{e,i} = \hat{y}_i \\ 2 \cdot \frac{\|\mathbf{a}_{e,i} - \mathbf{H}_{e, \mathcal{C}_{e, y_{e,i}}, \mathcal{C}_{e, \hat{y}_i}}\|_2}{\|\mathcal{C}_{e, y_{e,i}}\|_2 + \|\mathcal{C}_{e, \hat{y}_i}\|_2} & \text{if } y_{e,i} \neq \hat{y}_i, \end{cases} \quad (4)$$

where we scale the distance by the average norm of the two considered centroids.

We highlight that our hypothesis is similar but substantially different, from the one formulated in LfF [30]: here, we are free from the assumption bias features are learned earlier than all the more robust ones by the models, and even more from the assumption that they are the very first features learned by the models. We only state that at the point when the model learns these features, the misclassified elements will mainly be bias-conflicting: we can extract this information by monitoring the bias-target misalignments.

3.4 Debiasing the model

Once having extracted the bias alignment labels for each sample, we can start training and debiasing our actual model. In this work, we propose unbiased models having the same architecture and learning parameters as the bias extractor one, although no explicit constraint forbids us to use different models. Our approach here consists of modifying the objective function optimized during the training of the model with two goals: increasing the weight of the bias-misaligned samples (as they contain relevant information on generalized n compared to other representatives of the same class) and moving these samples away from the bias centroid that tends to attract them.

3.4.1 Loss function reweighting

As shown by LfF [30] and other works like [21], reweighting the loss function to up-weight the bias-conflicting elements is an efficient method to orient the training in a less biased direction as the correlation bias-target will be less emphasized in the loss. In this work, we assign the weight r_i to the i -th sample according to

$$r_i = \begin{cases} \frac{1}{\rho_{\hat{b}_i}} & \text{if } x_i \in \mathcal{D}^{\parallel} \\ \frac{1}{1 - \rho_{\hat{b}_i}} & \text{if } x_i \in \mathcal{D}^{\perp}, \end{cases} \quad (5)$$

where we define

$$\rho_c = \frac{\|\mathcal{D}_c^{\parallel}\|_0}{\|\mathcal{D}_c\|_0}. \quad (6)$$

In a nutshell, misaligned samples receive a weight that is proportionally inverse to their cardinality in the c -th class. This has the effect of strongly encouraging the learning of bias-misaligned samples, over bias-aligned ones. In the feature space, we can interpret the resulting reweighted loss $\mathcal{L}(y_i, \hat{y}_i, r_i)$ as an *attractive force* on the bias-misaligned samples, that are being pulled toward their class centroids.

3.4.2 Bias alignment information removal

Besides having a loss reweighting to favor misaligned sample learning, we can also discourage the model from learning any information related to bias alignment to the target. To estimate how much of this information is learned by the model, at the bottleneck we plug an auxiliary classification head that we call *information removal head* (IRH). This head is trained to minimize a cross-entropy loss $\mathcal{L}(b_i, \hat{b}_i)$. Its performance is an important indicator for us, as reveals how much the latent space is similar (or different) from the vanilla bias-capturing model, when it was the most fitted to the bias (at epoch e^*), from the bias-target misalignment perspective. To be more accurate, this is estimated through the computation of the mutual information between b_i and \hat{b}_i under the bias misalignment condition

$$\mathcal{I}^{\perp} = \sum_{j,k} p_{b,\hat{b}}^{\perp}(j,k) \log \left[\frac{p_{b,\hat{b}}^{\perp}(j,k)}{p_b^{\perp}(j)p_{\hat{b}}^{\perp}(k)} \right], \quad (7)$$

where

$$p_{b,\hat{b}}^{\perp}(j,k) = \frac{\sum_i b_i \cdot \delta_{\arg\max(b_i),j} \cdot \delta_{\hat{b}_i,k}}{\|\mathcal{D}^{\perp}\|_0} \quad (8)$$

is the joint probability between b and \hat{b} calculated on the bias-target misaligned samples, $\delta_{i,j}$ is the Kronecker delta, and p_b^{\perp} , $p_{\hat{b}}^{\perp}$ are the two marginals. (7) is differentiable, as b_i is the softmax-ed output of the IRH: hence, we are allowed to minimize this term, eventually scaled by a hyper-parameter $\lambda_{\mathcal{I}^{\perp}}$.

As also displayed in Fig. 2, the mutual information does not contribute to the information removal head's update, but it is propagated directly back to the backbone. Minimizing the information over the bias-target misaligned samples can be seen as a *repulsive force*, and should not be performed also on the bias-target aligned ones: indeed, we do not wish to destroy information related to the target class but to remove the link between bias misaligned samples and their attractor class. In the next section, we will test our approach and compare it with other state-of-the-art methods.

4 Experiments

Every result here presented is averaged over three seeds as done in most of the literature, and every algorithm is implemented in Python, using PyTorch 1.13, and trained on GPUs Nvidia GeForce RTX3090 Ti equipped with 24GB RAM. As we compare our method to the other unsupervised state-of-the-art methods, the best-unsupervised accuracies are systematically in bold and the second best are underlined. Besides, we also highlight in red the best method overall to see how our method compares even to the supervised ones. The source code is available at https://github.com/renahon/mining_bias_target_alignment_from_voronoi_cells/.

4.1 Datasets

Here below we describe at a glance the various datasets employed for the quantitative evaluation of our method.

Biased MNIST. The first dataset we are using is Biased MNIST, which was first introduced by Bahng *et al.* [4]. The 60k samples of this dataset consist of a colored version of the famous handwritten digits dataset MNIST with some correlation ρ between the color and the digits. To build it, first one specific color gets assigned to each of the ten digits; then each of the samples gets its background color. We test four levels of color-digit correlation ρ : 0.99, 0.995, 0.997, and 0.999. The effect of the bias (namely, the background color) is evaluated by testing the model on a completely unbiased dataset, with $\rho = 0.1$.



Figure 4: Example of bias-aligned (top row) and bias-misaligned (bottom row) samples (of target values 3, 5, and 7) for the Biased-MNIST dataset [4]

In fig.4, we can see an example of samples from this dataset that are either bias-aligned (top-row) or bias-misaligned (bottom-row). We can see on the top aligned row that each iteration of the same digit has the same color and that in the bottom misaligned one, the same digit has never that color. For example, all threes are yellow in the bias-aligned row whereas none of them is in the bottom row.

Multi-Color MNIST. Building on top of Biased MNIST, Li *et al.* proposed in [27] a bi-colored version to better benchmark the performance of current models on multiple biases at once. Here, the left and the right side of the background have two different colors, with a correlation to the target ρ_L and ρ_R . We follow their proposed setup, with $(\rho_L, \rho_R) = (0.99, 0.95)$.

CelebA. CelebA [28] is a real-world dataset commonly used to test debiasing performance. It is a face classification dataset provided with 40 attributes for each of the 203k image samples. The task we solve here is to classify “blond” or “not blond” hair, with the main bias lying on the gender, as the dataset presents a natural bias for “females” to have the “blond” attribute.

9-class ImageNet. The 9-class ImageNet dataset was proposed by [4], consisting of the extraction of a subset of 9 super-classes from ImageNet-1k, balanced to have each class correlated to a specific texture bias.

ImageNet-A. ImageNet-A was proposed by Hendrycks *et al.* in [19] as a subset of cropped images from ImageNet, purposely selected to be very hard to be classified by state-of-the-art CNNs. They were precisely selected among the subset misclassified by a cluster of ResNet50 models. Following [4, 20, 5], we use it to test our performance when training on 9-class ImageNet.

4.2 Model architecture and training details

For our experiments, we systematically used the same architecture for our vanilla model that allows us to measure the distances to decision hyperplanes and the model to debias. Therefore the architecture details are given below for each dataset stand for both models. Regarding the information removal head, we used SGD optimization with a learning

rate of 0.1 for each model and dataset. Besides, we use $\lambda_{T^\perp} = 2$ for all our experiments.

Regarding the experiments on Biased MNIST, we used the same fully convolutional network used for ReBias [4] and Irene [33], of four convolutional layers with 7×7 kernels, with a batch normalization after each of these layers. Regarding training, we followed the implementation used in [33] of 80 epochs, with an initial learning rate of 0.1 decayed by 0.1 at epochs 40 and 60, and a weight decay of 10^{-4} . For Multi-Color MNIST, we employ as architecture the same 3-layer MLP used in [27], trained for 500 epochs (until the model starts overfitting on the training set), using the same optimization strategy as in [27]. For the experiments on CelebA and 9-class ImageNet, we used a pre-trained Resnet-18 and the same optimization strategy as in [30, 20].

4.3 Discussion

Here we compare our method to the current state-of-the-art, both supervised and unsupervised, on the multiple datasets presented in Sec. 4.1. In what follows, we divide the *supervision level* of the method into three categories: bias-agnostic, bias-aware (using an extra ground-truth bias label), and "bias tailored" (BT) where the method does not rely on bias labels, but by construction, it captures specific biases (like the texture [4, 37]).

Table 1: Results on Balanced Biased MNIST when training with different correlations color-digit ρ .

Method	Bias agnostic	Test accuracy [%] (\uparrow)			
		$\rho=0.999$	$\rho=0.997$	$\rho=0.995$	$\rho=0.99$
Vanilla	✓	11.2	40.5	72.4	88.4
Rubi [9]	✗	13.7	90.4	43.0	93.6
EnD [34]	✗	52.3	83.7	93.9	96.0
BCon+BBal [20]	✗	94.0	97.3	97.7	98.1
HEX [37]	BT	10.8	16.6	19.7	24.7
ReBias [4]	BT	26.5	65.8	75.4	88.4
LearnedMixin [12]	✓	12.1	50.2	78.2	88.3
LF [30]	✓	15.3	63.7	90.3	95.1
SoftCon [20]	✓	65.0	88.6	93.1	95.2
Ours	✓	58.7 \pm 21.8	92.7 \pm 1.2	95.5 \pm 0.8	97.7 \pm 0.4

Results on Biased MNIST. Our results on Biased MNIST, presented in Table 1, show that our method achieves state-of-the-art performance, even when compared with supervised ones, for not extreme values of ρ : the only method that yields better results on the three lower correlations levels is the use of the associated BiasContrastive and BiasBalanced losses [20]. On the unsupervised field, we get better accuracies than our competitors except for the highest correlation level. We can observe in this case a very high standard deviation because of the high stochastic noise of the gradient: the very few bias-target misaligned (60 in total, constituting the 0.1% of the train set) searches for the perfect moment to mark. We hypothesize these difficulties are caused by the large gradients, which make the bias-target information extraction noisy: we tried to perform the bias extraction for this specific setup having a smaller learning rate (0.01), and the performance improved to $72.6\% \pm 11.6$. Tuning properly the learning rate in extreme scenarios is a key element towards a successful bias-target alignment information extraction.

Table 2: Test accuracy on four subsets of Multi-Color MNIST. The "Unbiased" one is the average of the four.

Method	Bias agnostic	Test accuracy [%] (\uparrow)				Unbiased
		$A_{\text{left}} / A_{\text{right}}$	$A_{\text{left}} / C_{\text{right}}$	$C_{\text{left}} / A_{\text{right}}$	$C_{\text{left}} / C_{\text{right}}$	
Vanilla	✓	100.0	97.1	27.5	5.2	57.4
LF [30]	✓	99.6	4.7	98.6	5.1	52.0
EIIL [14]	✓	100.0	97.2	70.8	10.9	69.7
PGI [1]	✓	98.6	82.6	26.6	9.5	54.3
DebiAN [27]	✓	100.0	95.6	76.5	16.0	72.0
Ours	✓	100 \pm 0.0	90.9 \pm 3.5	77.5 \pm 2.8	24.1 \pm 1.8	73.1 \pm 0.9

Results on Multi-Color MNIST. The Multi-Color MNIST dataset [27] helps us to test the performance of our method on multiple biases at the same time. Considering that there is a distinct correlation between the digits and the left color and them and the right background color, four accuracies are measured, regarding whether each background color is bias-aligned (A_{left} for the left background color) or bias-conflicting (C_{right} for the right background one). The most complex setup is $C_{\text{left}} - C_{\text{right}}$: it is below random guess for the vanilla model, but also three out of five tested debiasing methods. The average of these four metrics constitutes “unbiased accuracy”. In contrast, every method reaches 100% accuracy or close on the $A_{\text{left}} - A_{\text{right}}$ configuration. Even in this case, our method achieves state-of-the-art results for this dataset. More specifically, we record the best unbiased accuracy, and we improve the best score on the double-conflicting setup by the +8%. For instance, LfF [30], whose main assumption is close to ours, while emphasizing the early choices of its bias-extracting model seems to perform very unevenly regarding the two biases (around 5% accuracy when for C_{right}). This further strengthens our choice of not extracting biased information, but the bias-target alignment, and waiting for the best learning moment to extract such information.

Table 3: Results on CelebA, targeting the attribute “blond”, with a bias towards gender.

Method	Bias agnostic	Test accuracy [%] (\uparrow)	
		Unbiased	Bias-Conflicting
Vanilla	✓	79.0	59.0
EnD [34]	✗	86.9	76.4
LNL [25]	✗	80.1	61.2
DI [38]	✗	90.9	86.3
BCon+BBal [20]	✗	91.4	87.2
Group DRO [32]	✓	85.4	83.4
LfF [30]	✓	84.2	81.2
Ours	✓	90.2\pm1.1	84.5\pm2.0

Results on CelebA. On the CelebA dataset, two test setups are employed: the “unbiased”, where the average of the scores obtained on each of the target-bias combinations (here blond-male, blond-female, not blond-male and not blond-female) is considered, and the “bias-conflicting” one, where the two bias-aligned combinations are not considered. On both the metrics, our approach ranks the best unsupervised (4.8% more on the unbiased metric), and the third overall.

Table 4: Test accuracy on 9-class ImageNet and ImageNet-A.

Method	Bias agnostic	Test accuracy [%] (\uparrow)	
		9-class ImageNet	ImageNet-A
Vanilla	✓	94.0	30.5
ReBias [4]	✗	94.0	30.5
StylImageNet [17]	BT	88.4	24.6
LearnedMixin [12]	BT	79.2	19.0
RUBi [9]	BT	93.9	31.0
LfF [30]	BT	91.2	29.4
SoftCon [20]	BT	95.3	34.1
FairKL [6]	BT	95.1	35.7
Ours (BagNet [8])	BT	96.4 \pm 0.0	34.5 \pm 3.4
Ours	✓	95.5 \pm0.2	34.2 \pm0.9

Results on 9-class ImageNet. When working at debiasing 9-class ImageNet, all other state-of-the-art methods become bias-tailored: indeed, they use BagNet18 [8] as a bias-extracting model for its known tendency to fit texture because of its small receptive fields (as 9-class ImageNet and ImageNet-A are known to be very biased towards texture), instead of ResNet18 which is the model we are trying to de-bias. If we want to perform bias-agnostic debiasing, we shouldn’t rely on that kind of bias-extracting model chosen to fit the bias type of the dataset. However, to compare our method to theirs on an equal footing we tested two configurations:

- ours (+BagNet), where we extract the bias-conflicting samples from training BagNet18 and then proceeded to debiasing ResNet18;
- ours, where we extract the bias-conflicting samples directly from the ResNet18, which makes us the only bias-agnostic method tested on this dataset.

We obtain comparable results to the state-of-the-art on ImageNet-A (respectively 1.2 and 1.5% below the best-performing method in FairKL [5]) and the two best results overall on 9-class ImageNet. Interestingly, we get state-of-the-art results with our method, when extracting information directly from ResNet18.

Table 5: Ablation study on Biased MNIST with $\rho = 0.99$.

Weighted \mathcal{L}	Information removal head	Misaligned only	Test accuracy [%](\uparrow)
			88.4 \pm 0.5
✓			95.5 \pm 0.5
✓	✓		97.2 \pm 0.4
✓	✓	✓	97.7 \pm0.4

Ablation study. We tested the effect of the different modules of our method on Biased MNIST, training with $\rho = 0.99$. The results in Table 5 show that the use of the reweighted loss function \mathcal{L} yields an average increase in accuracy of 7.1% and that the use of the information removal head (IRH) further increases our score of 1.7% more. Finally, employing a conditional mutual information term (on the bias-target misaligned elements only) in place of total information removal provides an extra gain in performance.

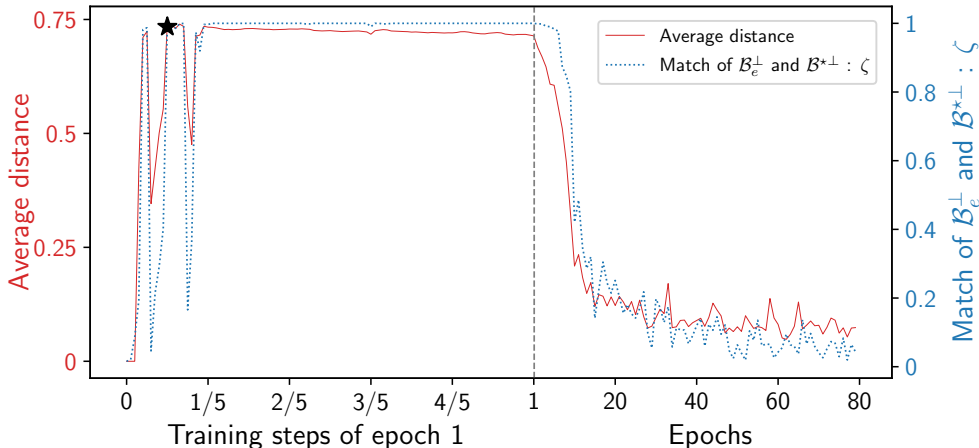


Figure 5: Evolution of the relative distance to the target Voronoi cell for the misclassified elements (red curve) and bias alignment information match with the ground truth \mathcal{B}^* (blue dashed line).

Relevance of the Voronoi distance metric. We also measured the average relative distances from the misclassified samples at each epoch, comparing it to the match of the vector of inferred bias labels $\mathcal{B}_e = (\text{argmax}(b_{e,i}))_i^n$ to the ideal ground truth of bias labels \mathcal{B}^* (which is provided in Biased MNIST). It’s computed over the set of misclassified samples \mathcal{D}^\perp as the bias-misaligned elements are our work’s main point of interest. The value represented by the blue dashed line represents therefore the bias alignment information match and is computed as:

$$\zeta = \frac{\sum_{i|x_i \in \mathcal{D}^\perp} \mathbb{1}_{\text{argmax}(b_{e,i})=b_i^*}}{|\mathcal{D}^\perp|} \quad (9)$$

As shown in Fig. 5, we can see a peak after the first few iterations, as the model fits the color bias with even less one epoch of training. We have marked with \star the peak over the average relative distances. As the training goes on, the average relative distance goes down as expected: the misclassified samples stay closer to the decision hyperplane. We

observe that the relative average distance is a good proxy for knowing when to learn the optimal bias alignment \mathcal{B}^* , as the two curves show a similar trend.

General discussion and limitations. Through the conducted experiments, we have observed that our method establishes, in most of the considered setups, a new state-of-the-art for bias-agnostic approaches, and in some cases even outperforms supervised methods, such as in 9-class ImageNet and the double-biased Multi-Color MNIST. A limitation of the proposed approach appears when the correlation between bias and target is extremely high ($\rho = 0.999$ in Biased MNIST). Since it heavily relies on the extraction of these bias-conflicting samples, when the stochastic noise overwhelms the extraction of the bias misalignment information, evidently the proposed method will be sub-optimal. A possible solution to this problem relies upon the use of a “sufficiently small” learning rate. Finally, our method strongly depends on the existence of these bias-conflicting elements: in a fully-biased dataset, where the alignment bias-target $\rho = 1$, since we have no information to extract from the training set, our approach is expected to fail.

5 Conclusion

In this paper, we presented an unsupervised, bias-agnostic debiasing approach, whose performance is typically in the same range as state-of-the-art supervised methods. We proposed a new bias-target alignment extraction method based on the distance between the misclassified samples and the closest Voronoi hyperplane separating them from their target class. Based on this distilled information, we proposed a debiasing method consisting of two synergetic elements. The first consists of a reweighted cross-entropy loss, where the weights of the samples reflect the bias-target (mis)alignment. The second is a bias-target misalignment information removal term, acting as a regularizer for the latent space. We tested our method on several debiasing benchmarks, recording a new state-of-the-art for unsupervised debiasing in most of the considered scenarios, although no specific hyper-parameters tuning has been performed. In extreme cases, where the bias-target alignment is extremely high, we have observed that the proper choice of the vanilla model’s learning setup is crucial for the success of the proposed approach, and its exploration is left as future work.

Acknowledgments

This work was granted access to the HPC resources of IDRIS under the allocation 20XX-AD011014080 made by GENCI.

References

- [1] Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron C. Courville. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2021. 3, 8
- [2] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellaker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings, 2018. 1
- [3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2019. 3
- [4] Hyojin Bahng, Sanghyuk Chun, Sangdoon Yun, Jaegul Choo, and Seong Joon Oh. Learning De-biased Representations with Biased Representations, June 2020. arXiv:1910.02806 [cs, stat]. 1, 3, 7, 8, 9
- [5] Carlo Alberto Barbano, Benoit Dufumier, Enzo Tartaglione, Marco Grangetto, and Pietro Gori. Unbiased Supervised Contrastive Learning, November 2022. arXiv:2211.05568 [cs, stat]. 1, 3, 7, 10
- [6] Carlo Alberto Barbano, Benoit Dufumier, Enzo Tartaglione, Marco Grangetto, and Pietro Gori. Unbiased supervised contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023. 9
- [7] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in AI, May 2020. 2
- [8] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet, 2019. 9
- [9] Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. RUBi: Reducing Unimodal Biases in Visual Question Answering, March 2020. arXiv:1906.10169 [cs]. 1, 8, 9
- [10] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation, 2018. 1
- [11] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194, 2020. 2

- [12] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases, September 2019. arXiv:1909.03683 [cs]. 1, 8, 9
- [13] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning, 2018. 2
- [14] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning, 2020. 3, 8
- [15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 214–226, New York, NY, USA, January 2012. Association for Computing Machinery. 2
- [16] Pratyush Garg, John Villasenor, and Virginia Foggo. Fairness Metrics: A Comparative Analysis, January 2020. arXiv:2001.07864 [cs]. 2
- [17] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *CoRR*, abs/1811.12231, 2018. 1, 2, 3, 9
- [18] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3323–3331, Red Hook, NY, USA, 2016. Curran Associates Inc. 2, 3
- [19] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples, 2019. 7
- [20] Youngkyu Hong and Eunho Yang. Unbiased Classification through Bias-Contrastive and Bias-Balanced Learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 26449–26461. Curran Associates, Inc., 2021. 7, 8, 9
- [21] F. Kamiran and T.G.K. Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012. 3, 6
- [22] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*, pages 924–929, 2012. 2
- [23] Jian Kang, Tiankai Xie, Xintao Wu, Ross Maciejewski, and Hanghang Tong. InfoFair: Information-Theoretic Intersectional Fairness, December 2022. arXiv:2105.11069 [cs, math, stat]. 2
- [24] Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. Content and style aware generation of text-line images for handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(12):8846–8860, 2022. 2
- [25] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data, 2018. 1, 9
- [26] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jiheon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation, 2021. 1, 3
- [27] Zhiheng Li, Anthony Hoogs, and Chenliang Xu. Discover and Mitigate Unknown Biases with Debiasing Alternate Networks, September 2022. arXiv:2207.10077 [cs]. 3, 7, 8, 9
- [28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild, 2014. 7
- [29] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms, 2009. 1
- [30] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from Failure: De-biasing Classifier from Biased Classifier. In *Advances in Neural Information Processing Systems*, volume 33, pages 20673–20684. Curran Associates, Inc., 2020. 1, 3, 5, 6, 8, 9
- [31] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 560–568, New York, NY, USA, 2008. Association for Computing Machinery. 3
- [32] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization, April 2020. arXiv:1911.08731 [cs, stat]. 9
- [33] Enzo Tartaglione. Information Removal at the bottleneck in Deep Neural Networks, September 2022. arXiv:2210.00891 [cs]. 8
- [34] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13503–13512, 2021. 1, 3, 8, 9
- [35] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528, 2011. 1
- [36] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance, 2014. 1

- [37] Haohan Wang, Zexue He, Zachary C. Lipton, and Eric P. Xing. 1, 3, 8
- [38] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation, 2019. 9
- [39] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey, 2021. 1
- [40] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8792–8802, Red Hook, NY, USA, 2018. Curran Associates Inc. 1