Learning unbiased classifiers from biased data with meta-learning

Ruggero Ragonesi[1] Pietro Morerio[1] Vittorio Murino[1,2]
[1]Pattern Analysis and Computer Vision (PAVIS) - Istituto Italiano di Tecnologia
[2] Department of Computer Science, University of Verona
name.surname@iit.it

Abstract

It is well known that large deep architectures are powerful models when adequately trained, but may exhibit undesirable behavior leading to confident incorrect predictions, even when evaluated on slightly different test examples. Test data characterized by distribution shifts (from training data distribution), outliers, and adversarial samples are among the types of data affected by this problem. This situation worsens whenever data are biased, meaning that predictions are mostly based on spurious correlations present in the data. Unfortunately, since such correlations occur in the most of data, a model is prevented from correctly generalizing the considered classes. In this work, we tackle this problem from a meta-learning perspective. Considering the dataset as composed of unknown biased and unbiased samples, we first identify these two subsets by a pseudo-labeling algorithm, even if coarsely. Subsequently, we apply a bi-level optimization algorithm in which, in the inner loop, we look for the best parameters guiding the training of the two subsets, while in the outer loop, we train the final model taking benefit from augmented data generated using Mixup. Properly tuning the contributions of biased and unbiased data, together with the regularization introduced by the mixed data has proved to be an effective training strategy to learn unbiased models, showing superior generalization capabilities. Experimental results on synthetically and realistically biased datasets surpass state-of-the-art performance, as compared to existing methods.

1. Introduction

In classification tasks, it is widely recognized that deep learning architectures can learn large amount of data, reaching unprecedented outstanding performance. However, such models are also very sensitive to data, meaning that they are prone to errors with high confidence whenever test samples are drawn from a distribution different from that of the training set. One reason is that, in certain conditions, these models have problems to generalize well the classes considered as they likely memorize the training data rather than learning the salient characteristics of each category of examples. This behavior is especially evident when training data are biased, i.e., samples include spurious correlations with class labels or, in other words, the trained model learns some "shortcuts" to classify data, so failing to generalize the class properly [5,9]. For example, a fish can be classified as such due to the presence of the blue sea in which fishes are typically depicted, and not for the actual fish structure and appearance, hence a model may likely fail in case the input image depicts a fish located on a brown market table. Such shortcuts are learnt since most of the samples are characterized by a bias (fishes in the sea) while only a few samples are unbiased (fishes in unusual contexts), which prevents from proper generalization.

When optimizing models under the presence of biased data, the ground-truth knowledge of the bias is typically beneficial. For instance, having an additional annotation regarding whether the fish is in the sea or not can be used to drive the optimization towards a data representation invariant to such attribute (See Figure 1(a)). Several methods approached the problem in this way and sought for a data representation invariant to a known factor [1,2,6,12,20,26,27,29]: we term this problem *supervised debiasing*, i.e. the knowledge of the bias acts as an auxiliary data annotation that can be useful to consider in training in order to get invariance with respect to it. However, the hypothesis of having an additional label is unrealistic in most practical scenarios as it requires great effort during data annotation, and in some cases can even be impossible whenever the control of data gathering is unfeasible, hence the urge of methods that can generalize even without this additional supervision.

For these reasons, we face here the more challenging setting of the *unsupervised debiasing* problem: assuming that the ground-truth knowledge of the bias is not readily available, we attempt to (implicitly) infer this information while debiasing our model and achieving a successful generalization on the test set (See Figure 1(b)).

In this paper, we devised a two-stage algorithm tackling the unsupervised debiasing problem. First, we separate bi-

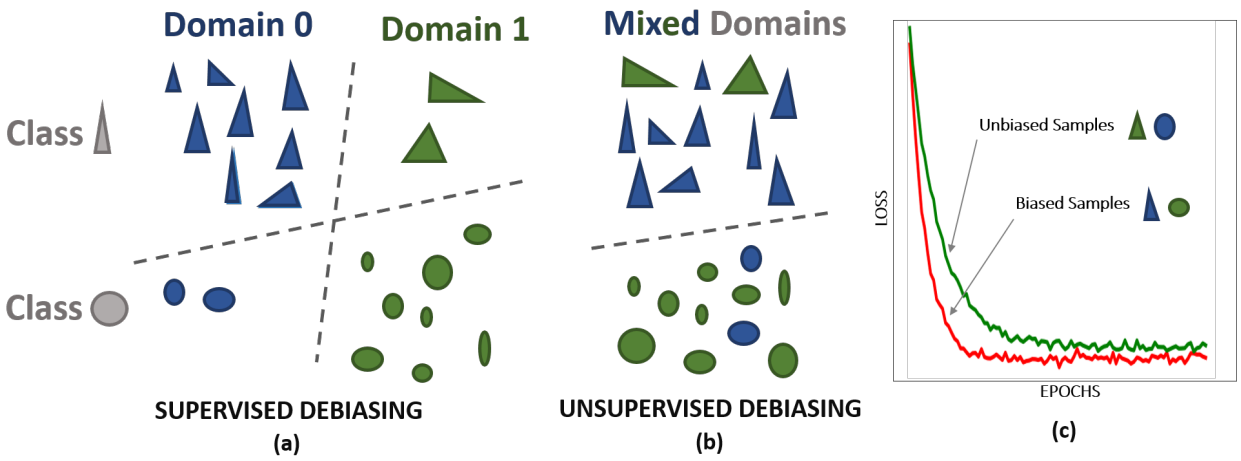

Figure 1. **Problem description.** (a) Biased dataset occurs when there is an imbalance regime regarding pairs (class, domain), where each class is observed mostly under one distribution, leaving other options under-represented. This results in trained models which do not generalize well. In the case of supervised debiasing case, one has additional annotations regarding the domain distribution. (b) In the unsupervised debiasing case, one has only access to the class labels. A possible approach to distinguish biased/unbiased samples is via pseudo-labeling. (c) The plots show that the loss for the biased samples are decreasing much faster than the loss for unbiased samples, proving that the former can be learnt more easily than the latter. (Best viewed in color).

ased from unbiased samples through a pseudo-labeling approach. Second, equipped with such (noisy) pseudo-labels, we manage the problem of learning from this data using a Meta-Learning approach (inspired by Finn et al. [8]) to produce data representation that can accommodate both biased and unbiased samples. While existing methods [6, 25] focused primarily on increasing performance on unbiased samples, overlooking the need for keeping high accuracy on biased samples as well, we aim instead at achieving high accuracy over both types of data.

To this end, grounding on the intuition that meta-learning is a suitable approach to learn effectively different tasks, we propose to treat the learning from biased and unbiased data as different (meta-)training tasks, followed by a meta-validation step devoted to produce data representations which are suitable for both, aimed at better generalization. For the latter, we generate (augment) new data by linearly interpolating [32] biased and unbiased samples, so producing samples which are more "neutral" than the original biased and unbiased images, so reducing the contribution of spurious correlations in the prediction (See Fig. 1(c)) and overall regularizing the training. We can generate more "neutral" representations by mixing biased and unbiased samples even if they are not perfectly subdivided by the initial pseudo-labeling stage. In other words, the method is robust to some level of contamination between the estimated biased and unbiased subsets. Interestingly, this is a notable characteristic of our approach making it suitable in realistic scenarios. In fact, we do not need to perfectly identify the biased/unbiased samples, nor knowing or determining the bias affecting the data: the splitting performed by any pseudo-labeling algorithm can be managed by the subsequent meta-learning and data augmentation stage, to regularize the training.

We validate our method on several benchmarks that are both synthetic with controlled bias (colored MNIST and Corrupted CIFAR-10) and more realistic (Waterbirds and BAR), showing outstanding performance as compared with existing methods.
To recap, the contributions of our work are:

- We face the challenging unsupervised debiasing problem by introducing a two-stage approach that, after the initial coarse identification of the biased and unbiased samples, can modulate the contribution of each example during the model training by a meta-learning strategy.
- Specifically, our novel approach considers learning from biased and unbiased samples as separate meta-training tasks, while *generating* new data by augmentation, managed as a (meta-)validation task. By jointly optimizing the original meta-training and the meta-validation tasks, we inject a strong regularization in the training process, so compensating the imbalance problem between biased/unbiased samples by neutralizing the bias, and ultimately leading to more general representation learning.
- Our approach, validated on datasets with controlled bias and realistic benchmarks, showed to outperform state-of-the-art performance by a significant margin.

The rest of the paper is organized as follows. In Section 2, we describe the related literature, highlighting the original aspects introduced. Section 3 reports our method, where we detail our two-stage approach. Section 4 presents the results and a thorough ablation analysis. Section 5 wrap-ups the work and sketches future research directions.

2. Related Works

Learning from biased data can be seen as a specific case of Out-Of-Distribution (OOD) domain generalization. This topic has been addressed with different methodologies, including meta-learning. Here, we briefly review the most related literature.

Learning from biased data. The problem of learning from biased data has been explored in past years in the supervised debiasing setting, i.e. when labels for the factor (bias) to be removed are available. Several methods approached the problem seeking an invariant data representation to a known factor. Such approaches rely on adversarial learning [1, 12, 31], variational inference [7, 23, 24], Information Theory [26], re-sampling strategies [20], or robust optimization [27]. Invariant Risk Minimization [2] seeks an optimal representation which is invariant across domains while EnD [28] attempts to disentangle useful information from biased in the data features.

Few recent works [3, 18, 22, 25] have addressed the unsupervised debiasing problem. [3] formalizes the *cross-bias* problem where malicious shortcuts exist, easing the fit of training data, whereas the same shortcuts result harmful at inference stage. The solution is learning a debiased model which is statistically independent from the one computed by a parallel computational stream that is guaranteed to be affected by the bias by design. In [25], the nature of the aforementioned "shortcuts" is analyzed in terms of fitting speed at training time. Nam et al. show that biased samples are learnt faster than the unbiased ones. The relative difficulty of each sample is cast into a weight that modulates its learning rate: in this way, at training time, it is given more importance to the few outlying samples that do not follow the shortcuts. To this end, an ensemble of networks is trained, similarly to [3]. [18] tackles the problem via robust optimization, considering a worst case loss of a subpopulation of the dataset (typically samples with the highest loss). In [13, 22], the training data is split in two subsets, relying on the predictions of a baseline model. In [22], the most difficult samples (likely those that do not follow shortcuts) are then upsampled. [13] identifies patches from the two splits and then swaps them in order to produce additional samples with which to train the debiased model.

Our work does not rely on an ensemble of networks to have a reference biased model. Instead, we perform a pseudo-labeling approach to split the dataset in two subsets and then treat them as two separate tasks to be learned via meta-learning. We also avoid data upsampling as in [22] and [20], whereas we pursue a data augmentation approach to combine biased and unbiased samples. Inspired by Mixup [32], we mix factors which are peculiar of the bias regime (likely representing a shortcut to infer the class) with those that do not follow such rules. The newly generated samples are expected to break the spurious correlations that affect the original data.

Meta-Learning for Out-Of-Distribution domain generalization. A class of meta-learning methods based on bi-level optimization (e.g., Model Agnostic Meta-Learning [8]), relies on an inner-loop stage optimizing model's meta-parameters on source data, and an outer-loop stage that updates the model parameters on (meta-)validation data. This nested optimization which involves computing a gradient through a gradient, has been shown to be effective for a fast adaptation of the model to the validation data. The goal is learning from an (empirical) training task distribution so to generalize and learn faster (i.e., with fewer samples) the validation task. Subsequently, other methods have tackled the problem of Domain generalization (DG) ([4, 19, 21], to cite a few), casting the problem of learning from multiple tasks to learning from multiple distributions/domains.

We adopt the same general scheme, however we face a considerably distinct problem: while in DG, different domains are fairly balanced, we deal with a severe data imbalance, that is, biased vs. unbiased, seen here as domains. This domain data imbalance is so dramatic that the model likely learns domain attributes to perform inference, hampering its generalization capabilities. This requires a tailored solution that we found effective through data augmentation, in order to attempt to reduce the imbalance problem. Moreover, differently from previous methods that rely on multiple source domains, we relax the hypothesis of having domain labels and adopt a pseudo-labeling approach to overcome this issue.

3. The Method

We consider supervised classification problems with a training set $\mathcal{D}_{train} = \{x_k, y_k, d_k\}_{k=1}^{N}$, where x_k are raw input data, y_k class labels and d_k domain labels. In the case of a biased dataset, $\mathcal{D}_{train}$ has several classes $y^i, i = 1, ..., C$, which are considered to be observed under different domains $d^j, j = 1, ..., D$; D can be different from C but here, for clarity and without losing generality, we consider the case of $D = C$. When the majority of samples of a specific class y^i is observed under a single domain d^j, while other domains are under represented in the dataset, we say that the pair (y^i, d^j) is biased, i.e. there is a spurious correlation between class and domain.

We define $\mathcal{D}_{bias}$ as the subset of training samples that exhibit spurious correlations and $\mathcal{D}_{unbias}$ as the subset of samples with under represented pairs. Such subsets are highly imbalanced, i.e. $|\mathcal{D}_{bias}| >> |\mathcal{D}_{unbias}|$. For instance, in a cats vs. dogs classification problem, most of the cats may be observed in an indoor home environment, while most of the dogs may be observed in outdoor scenes. For both classes, very few images are outside of the main distribution.

We aim to tackle the *unsupervised debiasing* problem, which means that we do not have access to domain la-

bels d nor to other bias information, hence we consider a training set containing only input data and class label, $\mathcal{D} = \{x_k, y_k\}_{k=1}^{N}$. We want to train a neural network f_θ on $\mathcal{D}$, with parameters θ, to be deployed on test data $\mathcal{D}_{test}$ not seen during training. θ are usually found via Empirical Risk Minimization (ERM), i.e. minimizing the expected Cross-Entropy loss over the training data:

$$\begin{aligned} \min_{\theta} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}} \mathcal{L}(\mathcal{D}, f_\theta) \\ \mathcal{L}(\mathcal{D}, f_\theta) = \mathbf{y}^T \log(\sigma(f_\theta(\mathbf{x}))) \end{aligned} \tag{1}$$

where σ is the softmax function. In such scenario, when trained via ERM, a model focuses mostly on the more numerous biased samples, underfitting the unbiased ones: this results in a biased model that uses spurious correlation (e.g., background) as a possible way to make inference, instead of correctly learning the class semantic. In general, $\mathcal{D}_{test}$ follows a data distribution different from $\mathcal{D}_{train}$, i.e. the biased pairs may be not the majority of samples. Hence it is important to have a model that can be deployed on both biased and unbiased pairs.

Our method tackles the unsupervised debiasing problem with a two-stage approach. In the first stage, we separate biased from unbiased samples through a pseudo-labeling algorithm. Equipped with such pseudo-labels, we train a model to produce a data representation that can accommodate both biased and unbiased samples. In the following, we detail the two main stages of our method.

3.1. Bias Identification

In this stage, our goal is to split the training set $\mathcal{D}$ into two disjoint subsets $\hat{\mathcal{D}}_{bias}$ and $\hat{\mathcal{D}}_{unbias}$ that should resemble the actual, ground-truth $\mathcal{D}_{bias}$ and $\mathcal{D}_{unbias}$. In [25], it is shown how the biased samples are learnt faster than the unbiased ones: the imbalanced nature of the dataset makes the model more prone to learn first the numerous biased samples and later those unbiased. This behaviour can be observed by looking at the loss function trends of the two subsets (See Fig. 1(c)). We exploit the fact that samples from D_{bias} are easily learnt during training, to design a strategy for splitting the dataset. We train a neural network f_ϕ via ERM until it reaches a training accuracy of γ, where γ is a hyper-parameter denoting the target accuracy. When the model reaches the desired accuracy level, the training stops and a forward pass of the entire training set is performed. Now, samples that are correctly predicted are assigned to $\hat{\mathcal{D}}_{bias}$ while those not correctly predicted are assigned to $\hat{\mathcal{D}}_{unbias}$. More formally:

$$\begin{aligned} \hat{\mathcal{D}}^{\gamma}_{bias} = \{(x, y) \in \mathcal{D} \mid \sigma(f^{\gamma}_{\phi}(x)) = y\} \\ \hat{\mathcal{D}}^{\gamma}_{unbias} = \{(x, y) \in \mathcal{D} \mid \sigma(f^{\gamma}_{\phi}(x)) \neq y\} \end{aligned} \tag{2}$$

Using γ as hyper-parameter is convenient for two reasons. First, our setting of the amount of desired accuracy is dataset agnostic. This is different from prior work [22] that employs a similar strategy, but with the hyper-parameter controlling the number of epochs to train the model: in that case, the number of epochs are strictly dependent on the dataset that the model is trained on. Second, we can have a precise control of the amount of samples assigned to the two splits, e.g. $\gamma = 0.85$ implies that 85% of training data are assigned to $\hat{\mathcal{D}}_{bias}$ and 15% to $\hat{\mathcal{D}}_{unbias}$. In real use cases, we do not know the correct assignments of the samples to the splits, so we have to rely on a priori setting of this parameter.

3.2. Bias-invariant representation learning

Provided with pseudo-labels for the two estimated subsets $\hat{\mathcal{D}}_{bias}$ and $\hat{\mathcal{D}}_{unbias}$, we deal with the problem of learning data representations that are not only good for the biased data but can generalize well to unbiased samples. We adopt a neural network f_θ, trained from scratch, and we designed a bi-level optimization algorithm inspired by meta-learning to learn efficiently from such data.

Inner loop step. This is a meta-training step where we seek the best parameters θ for the two subsets $\hat{\mathcal{D}}_{bias}$ and $\hat{\mathcal{D}}_{unbias}$ via gradient descent:

$$\theta^* = \theta - \eta \, \nabla_\theta \left[(1-\gamma) \, \mathcal{L}(\hat{\mathcal{D}}_{bias}, f_\theta) + \gamma \, \mathcal{L}(\hat{\mathcal{D}}_{unbias}, f_\theta) \right] \tag{3}$$

where η is the learning rate. In this step, the two splits of the training data are treated as two separate tasks: we scale the two loss functions with two coefficients to deal with data imbalance ($|\hat{\mathcal{D}}_{bias}| >> |\hat{\mathcal{D}}_{unbias}|$). To rebalance the contributions from the two splits, an obvious choice is to set weights inversely proportional to the cardinality of the two subsets, which is nothing else than the fixed and controllable hyper-parameter γ.

Outer loop step. Standard meta-learning usually optimizes for the meta-test task using the parameters found in the inner loop, relying on a (typically small and clean) validation set. Here, we get rid of this assumption since do not have access to any held-out nor clean data, therefore we opt for a data augmentation approach in order to provide unseen data to the model.

We seek a representation that can conciliate both biased and unbiased samples and at the same time prevent the model from overfitting the meta-training data (the two subsets $\hat{\mathcal{D}}_{bias}$ and $\hat{\mathcal{D}}_{unbias}$), which is a common problem in meta-learning. We take inspiration from Mixup [32] as a way to combine samples from the two subsets. Mixup provides a convex combination of both input samples and labels and it has demonstrated its efficacy as an effective regularizer. Specifically, we feed the model with samples resulting from the mix of examples from biased and unbiased data, aiming at likely breaking the shortcuts present in the dataset (see Fig. 2).

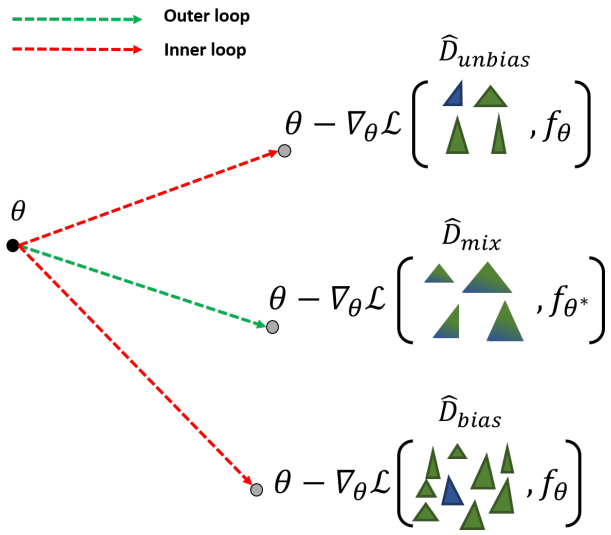

Figure 2. Starting from the current parameter configuration θ, gradients on $\mathcal{L}(\hat{\mathcal{D}}_{bias}, f_\theta)$ and $\mathcal{L}(\hat{\mathcal{D}}_{unbias}, f_\theta)$ are evaluated to produce the new configuration θ^*. The regularization step using mixed data aims at producing a contribution that decreases the loss function on $\hat{\mathcal{D}}_{bias}$, $\hat{\mathcal{D}}_{unbias}$, and $\hat{\mathcal{D}}_{mix}$, simultaneously, the latter estimated over the configuration θ^*. (Best viewed in color)

We construct $\hat{\mathcal{D}}_{mix}$ by mixing samples of $\hat{\mathcal{D}}_{bias}$, $\hat{\mathcal{D}}_{unbias}$, sampling the parameter $\lambda \sim Beta(\alpha, \beta)$:

$$\begin{aligned} x_{mix} &= \lambda\, \hat{x}_1 + (1-\lambda)\, \hat{x}_2 \\ y_{mix} &= \lambda\, \hat{y}_1 + (1-\lambda)\, \hat{y}_2 \\ (\hat{x}_1, \hat{y}_1) \in \hat{\mathcal{D}}_{bias}\ &, (\hat{x}_2, \hat{y}_2) \in \hat{\mathcal{D}}_{unbias} \end{aligned} \tag{4}$$

Computed the augmented samples x_{mix}, y_{mix}, the model is updated in the outer loop:

$$\begin{aligned} \mathcal{L} := \underbrace{(1-\gamma)\, \mathcal{L}(\hat{\mathcal{D}}_{bias}, f_\theta) + \gamma\, \mathcal{L}(\hat{\mathcal{D}}_{unbias}, f_\theta)}_{\text{Weighted ERM}} \\ + \zeta\, \underbrace{\mathcal{L}(\hat{\mathcal{D}}_{mix}, f_{\theta^*})}_{\text{Regularizer}} \end{aligned} \tag{5}$$

where ζ is a hyper-parameter controlling the regularization. Note that the first two losses are evaluated on the current parameters configuration θ, while the loss over the augmented samples is evaluated in the meta-state θ^* (see Eq. 3). This implies that the model has to compute a gradient through a gradient, similarly to what happens in optimization-based meta-learning methods. The hyperparameter ζ controls the amount of regularization in the final loss: if $\zeta = 0$, the method corresponds to a (weighted) ERM in which the contributions of the losses on the two subsets are scaled by $(1-\gamma)$ and γ. When $\zeta > 0$ the weighted ERM optimization trajectory is corrected by the regularization term. This corresponds to find parameters θ that are good for both $\hat{\mathcal{D}}_{bias}$ and $\hat{\mathcal{D}}_{unbias}$, but can also possibly reduce the loss value on the newly generated data samples

Algorithm 1 Learning to learn unbiased representations

Input: Dataset $\mathcal{D}$, initialized weights θ_0, learning rate η, hyper-parameters ζ, γ, T.
Output: learned weights θ
Initialize: $\theta \leftarrow \theta_0$
Identify $\hat{\mathcal{D}}_{bias}$ and $\hat{\mathcal{D}}_{unbias}$ by pseudo-labeling (Eq. 2)
for t $= 1, ..., T$ **do**
  Sample $(\mathbf{x_b}, \mathbf{y_b})$, $(\mathbf{x_u}, \mathbf{y_u})$ from $\hat{\mathcal{D}}_{bias}$, $\hat{\mathcal{D}}_{unbias}$
  Compute θ^* (Eq. 3) ▷ Inner loop step
  Sample $(\hat{\mathbf{x}}_\mathbf{1}, \hat{\mathbf{y}}_\mathbf{1})$, $(\hat{\mathbf{x}}_\mathbf{2}, \hat{\mathbf{y}}_\mathbf{2})$ from $\hat{\mathcal{D}}_{bias}$, $\hat{\mathcal{D}}_{unbias}$
  Construct $\hat{\mathcal{D}}_{mix}$ (Eq. 4) ▷ Produce augmented samples
  Update θ (Eq. 5) ▷ Outer loop step

$\hat{\mathcal{D}}_{mix}$. Accuracy is not so affected by the choice of the ζ value: indeed, it increases as long as this ζ assumes positive values up to reaching high performance quite steadily, after that the contribution of the regularization becomes too strong and accuracy decreases. We set the value of ζ to a fixed value ($= 10$) for all experiments. Further analysis on this parameter is reported in the Supplementary Material. The complete method is summarized in Algorithm 1.

4. Experiments

We show the effectiveness of models trained by our method in a series of benchmarks, ranging from toy problems with synthetic biases to more realistic image classification tasks. We compare with methods that tackle the same bias problem in both supervised and unsupervised way.

4.1. Benchmarks

Synthetic bias datasets. To control the bias in the data and for the sake of comparison, we adopt two benchmarks that have been employed by Nam et al. [25], namely Colored MNIST and Corrupted CIFAR-10[1,2]. The first is a modified version of the standard digit recognition dataset [17], in which colors are added in order to artificially induce a bias in the dataset. The dataset is made of 60,000 training RGB images and 10 classes. Each sample is colored with a color tone which is randomly sampled from a Gaussian distribution whose mean is specific for each class; in practice, each class in the training data is observed mostly under a certain color tone, while the test set has no specific correlation between classes and colors and is balanced. Corrupted CIFAR-10 is a modification of the original dataset [15] that has been introduced in [11]. There are 50,000 training RGB images and 10 classes. The bias here stems from the fact that each image is corrupted with a specific noise (e.g., Gaussian blur, salt and pepper noise, etc.). Specifically, each class has a privileged type of noise under which it is observed during training (e.g., most of car images are corrupted with motion blur). There are two versions of

this dataset, namely Corrupted CIFAR-10[1] and Corrupted CIFAR-10[2], with different types of noise.

Realistic bias datasets. We tested our method on real image datasets, Waterbirds and Bias Action Recognition (BAR). Waterbirds has been introduced by [27] and combines bird photos from the Caltech-UCSD Birds-200-2011 (CUB) dataset [30] with background images from the Places dataset [33]. There are 4,795 training images and the goal is to distinguish two classes, namely *landbird* and *waterbird*. The bias here is represented by the background of the images: most landbirds are observed on a land background while most waterbirds are observed on a marine background. BAR has been introduced by [25] as a realistic benchmark to test model's debiasing capabilities. It is constructed using several data sources and consists of 1,941 photos of people performing several actions, and the task is to distinguish them in 6 classes: Climbing, Diving, Fishing, Racing, Throwing and Vaulting. The bias arises from the context in which action photos are observed at training: for instance, climbing actions are performed in a dry mountain scenario at training time, whereas in the test set, they are set in a snowy environment. For details, readers can refer to the original paper [25].

4.2. Performances

We report the performance of our approach on the different benchmarks, starting from the Colored MNIST and corrupted CIFAR-10 since they have controlled bias for which we can better discuss the results. Since we deal with biased training data and balanced data in testing, we report both accuracies on the testing subset of unbiased samples only (those under-represented in the training data) as well as over the entire test set (biased + unbiased), to assess how much we lose on the biased samples. In fact, as we learn features having higher generalization capacity, spurious correlations are likely less exploited to classify biased examples, and this may cause a drop in performance on such samples.

For Colored MNIST, our network f_θ is an MLP with 3 hidden layers with 100 neurons each. We used ResNet-18 [10] as a backbone for Corrupted CIFAR-10 and BAR, and ResNet-50 as backbone for Waterbirds. We remove the last layer from such backbones, adding a 2-layer MLP head on top of it. ResNet is pre-trained on ImageNet [16]. The meta-parameter θ^* is computed only for the last two fully connected layers while the backbone is trained with only the contribution of the weighted ERM in Eq. 5 ($\zeta = 0$). We set the learning rate $\eta = 0.001$ for all datasets with batch size of 256 on synthetic biased data and 128 for realistic bias data. We used Adam [14] as optimizer. All the experiments comply the same evaluation protocol used in the competing methods for a fair comparison. Implementation details are reported in the Supplementary Material.

Results for synthetic bias datasets. Tables 1 and 2 present the performance on synthetic biased datasets, reporting the overall average accuracy and the one for unbiased samples only, respectively. We compare against two baselines, a model trained by Empirical Risk Minimization (ERM) and our method with $\zeta = 0$, which cancels out the contribution of the regularization brought by the outer loop step in Eq. 5. This second baseline only weighs the contributions of the two splits found via pseudo-labeling. We also compare our approach with several former methods to learn unbiased representations, either using annotation for the bias or not. For the methods requiring explicit knowledge of the bias, we consider REPAIR [20], which does sample upweighting, and Group-DRO [27], which tackles the problem using robust optimization. We finally report the performance of Learning from Failure (LfF) [25], which learns a debiased model without exploiting the labeling of the bias.

We consider different ratios of the bias (ranging from 95% up to 99.5%) as in [25]. This ratio indicates the actual percentage of the dataset belonging to $\mathcal{D}_{bias}$ and $\mathcal{D}_{unbias}$. This ratio is also linked to the parameter γ used in the pseudo-labeling stage, but in actual use cases it is not known. Hence, in all experiments, we made an arbitrary, largely loose choice for it, and fix the hyper-parameter $\gamma = 0.85$, i.e. we consider 85% of the training data as biased, and then assigned to $\hat{\mathcal{D}}_{bias}$, and the remaining 15% to $\hat{\mathcal{D}}_{unbias}$. Since γ is a sensitive parameter, we provide an ablation analysis in which we show how the performance changes as γ varies (see Section 4.3 below). We set $\zeta = 10$ throughout all the experiments: in the Supplementary Material, we report an ablation about this parameter.

We observe consistent better results with respect to the competitors, for all datasets and all possible bias ratios. Interestingly, the difference from the baselines increases as the dataset is more biased (higher bias ratio): this indicates that our method is more effective as the bias is more severe. We note that these performances are reached starting from a very coarse split of the data (85/15%), while the actual biased/unbiased sets are much more corrupted (from 95/5% to 99.5/0.5%): this robustness towards the pseudo-labeling subdivision is particularly useful in actual scenarios where the bias ratio is unknown. We report an ablation study in this regard in Section 4.3. The weighted ERM ($\zeta = 0$) is already a strong baseline that surpasses, in some cases, former debiasing methods. Please, note that for both the unbiased samples and, in average, over the whole test set, the improvement is significant by a large margin. This shows that our method is not only better at generalizing over unbiased samples, but also maintains high accuracy on the biased set.

Results on the realistic biased datasets. In these trials, we still compare against the ERM baseline and Group DRO, as supervised method as before, and four unsupervised algorithms, LfF [25], CVaR DRO [18], ReBias [3], and JTT [22]. Performances are reported in Table 3. For

Dataset	Bias ratio	**ERM**	**REPAIR [20]**	**Group-DRO [27]**	**LfF [25]**	**Ours,** $\zeta = 0$	**Ours,** $\zeta = 10$
Colored-MNIST	95%	77.6 ± 0.44	82.5 ± 0.59	84.5 ± 0.46	85.3 ± 0.94	82.3 ± 0.99	**89.3** $\pm$ **1.02**
	98%	62.3 ± 1.47	72.9 ± 1.47	76.3 ± 1.53	80.5 ± 0.45	73.8 ± 0.87	**83.4** $\pm$ **0.97**
	99%	50.3 ± 0.16	67.3 ± 1.69	71.3 ± 1.76	74.0 ± 2.21	68.3 ± 0.98	**81.6** $\pm$ **0.96**
	99.5%	35.3 ± 0.13	56.4 ± 3.74	59.7 ± 2.73	63.4 ± 1.97	57.1 ± 1.05	**72.2** $\pm$ **0.87**
Corrupted CIFAR-10^1	95%	45.2 ± 0.22	48.7 ± 0.71	53.1 ± 0.53	59.9 ± 0.16	54.3 ± 1.40	**63.3** $\pm$ **1.11**
	98%	30.2 ± 0.77	37.9 ± 0.22	40.2 ± 0.23	49.4 ± 0.78	44.4 ± 0.90	**56.2** $\pm$ **0.89**
	99%	22.7 ± 0.97	32.4 ± 0.35	32.1 ± 0.83	41.4 ± 2.34	33.4 ± 0.91	**50.5** $\pm$ **0.98**
	99.5%	17.9 ± 0.86	26.3 ± 1.06	29.3 ± 0.11	31.7 ± 1.18	26.1 ± 0.94	**43.3** $\pm$ **0.97**
Corrupted CIFAR-10^2	95%	41.3 ± 0.46	54.1 ± 1.01	57.9 ± 0.31	58.6 ± 1.18	53.8 ± 1.21	**62.5** $\pm$ **0.91**
	98%	28.3 ± 0.77	44.2 ± 0.84	46.1 ± 1.11	48.7 ± 1.68	43.2 ± 0.96	**55.2** $\pm$ **0.98**
	99%	20.7 ± 0.81	38.4 ± 0.26	39.6 ± 1.04	41.3 ± 2.08	37.0 ± 0.99	**49.8** $\pm$ **1.01**
	99.5%	17.4 ± 0.85	31.0 ± 0.42	$.342 \pm 0.74$	34.1 ± 2.39	30.6 ± 0.89	**43.6** $\pm$ **1.32**

Table 1. **Accuracy on whole test set.** Accuracy (in %) evaluated on biased + unbiased test samples for different bias ratios. Best performance in bold.

Dataset	Bias ratio	**ERM**	**REPAIR [20]**	**Group-DRO [27]**	**LfF [25]**	**Ours,** $\zeta = 0$	**Ours,** $\zeta = 10$
Colored-MNIST	95%	75.2 ± 0.87	83.3 ± 1.23	83.1 ± 0.81	85.8 ± 0.66	82.1 ± 0.88	**89.2** $\pm$ **1.09**
	98%	58.1 ± 0.56	73.4 ± 0.79	74.3 ± 1.09	80.7 ± 0.56	73.3 ± 0.73	**83.4** $\pm$ **0.85**
	99%	44.8 ± 0.84	68.3 ± 0.75	69.6 ± 0.63	74.2 ± 1.94	67.6 ± 0.92	**81.6** $\pm$ **0.79**
	99.5%	28.1 ± 0.45	57.3 ± 0.61	57.1 ± 0.78	63.5 ± 1.94	56.8 ± 0.79	**72.1** $\pm$ **0.94**
Corrupted CIFAR-10^1	95%	39.4 ± 0.75	50.0 ± 0.89	49.0 ± 0.48	59.6 ± 0.03	54.3 ± 0.89	**63.3** $\pm$ **1.10**
	98%	22.6 ± 0.45	38.9 ± 0.64	35.1 ± 0.92	48.7 ± 0.70	44.1 ± 0.83	**56.1** $\pm$ **0.92**
	99%	14.2 ± 0.91	33.0 ± 0.57	28.0 ± 0.68	39.5 ± 2.56	32.3 ± 0.84	**49.6** $\pm$ **0.85**
	99.5%	10.5 ± 0.28	26.5 ± 0.46	24.4 ± 0.48	28.6 ± 1.25	25.6 ± 0.91	**42.1** $\pm$ **0.88**
Corrupted CIFAR-10^2	95%	34.9 ± 0.84	54.5 ± 1.04	54.6 ± 0.61	58.6 ± 1.04	53.6 ± 0.86	**62.3** $\pm$ **1.04**
	98%	20.5 ± 0.64	44.6 ± 0.83	42.7 ± 0.77	48.9 ± 1.61	43.8 ± 0.84	**55.5** $\pm$ **0.98**
	99%	12.1 ± 0.75	38.8 ± 0.75	37.1 ± 1.22	40.8 ± 2.06	36.4 ± 0.93	**49.7** $\pm$ **0.94**
	99.5%	10.0 ± 0.84	31.4 ± 0.53	30.9 ± 0.89	32.0 ± 2.51	29.8 ± 0.91	**43.0** $\pm$ **0.85**

Table 2. **Results on unbiased test samples.** Accuracy (in %) evaluated only on the unbiased samples for different bias ratios. Best performance in bold.

these datasets, we remind that we do not have the full control of the bias ratios. Specifically, in BAR we do not know exactly the biased/unbiased samples and, differently from Colored MNIST and Corrupted CIFAR-10, which have a balanced test set, Waterbirds test set is also imbalanced. In these cases, it is also important not only to cope with unbiased samples, but also to maintain accuracy on biased data. Hence, we aim here at finding a good trade-off between generalizing to unbiased samples while keeping high performance on biased data as well: that is why performances in Table 3 are reported as accuracies over both the entire test set (avg) and the unbiased samples for Waterbirds, and over the whole test set only (avg) for BAR. For Waterbirds, we note that we score favorably with respect to other unsupervised methods for unbiased sample subset: we reach comparable performance with JTT, but we outperform it on the whole test set. In other words, we are able to learn bias invariant representations without giving up accuracy on the biased samples. ERM and CVar DRO outperform ours as per the avg accuracy, but their accuracy drastically drops when considering unbiased samples only. We show also competitive performance against the supervised method Group DRO: without using any bias supervision, our method surpasses its average test accuracy even if the accuracy on biased data results lower (owing to the supervision in this case). Concerning the BAR dataset, since there is no ground-truth for the bias we report only the average accuracy over the whole test set: our method outperforms all other competitors by a considerable margin.

Method	**Bias supervision**	**Waterbirds**		**BAR**
		Acc. avg	Acc. unbias	Acc. avg
ERM	No	97.3%	72.6%	53.5%
CVaR DRO [18]	No	96.0%	75.9%	-
LfF [25]	No	91.2%	78.0%	62.9%
ReBias [3]	No	-	-	59.7%
JTT [22]	No	93.3%	86.7%	-
Ours ($\zeta = 10$)	No	94.1%	87.1%	64.3%
Group DRO [27]	Yes	93.5%	91.4%	-

Table 3. Performance on the whole (avg) and unbiased only test set: comparisons with baseline, unsupervised and supervised methods (see text).

	Bias Identification		Oracle		Random split	
Bias ratio	F1 score	Test Acc.(%)	F1 score	Test Acc.(%)	F1 score	Test Acc.(%)
95%	0.65	63.3	1.0	66.3	0.37	50.3
98%	0.62	56.2	1.0	59.4	0.34	40.4
99%	0.58	50.5	1.0	54.7	0.33	29.7
99.5%	0.54	43.3	1.0	49.0	0.32	21.5

Table 4. **Ablation study on bias identification.** F1 score of $\hat{\mathcal{D}}_{bias}$, $\hat{\mathcal{D}}_{unbias}$ compared to ground truth annotations $\mathcal{D}_{bias}$, $\mathcal{D}_{unbias}$ obtained with our Bias Identification strategy (Eq.2), using an oracle and with subsets generated randomly. We report the final test accuracy on the whole test set for the three different cases.

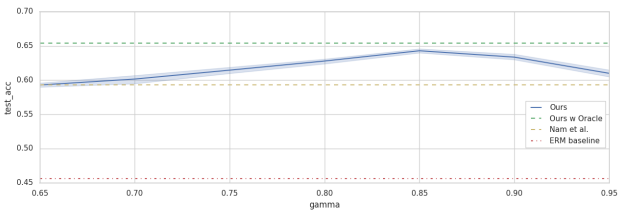

Figure 3. Test accuracies at different values of γ (from 0.65 to 0.95) averaged over 3 runs. We compare with ERM baseline, [25], and our method using the ground-truth bias knowledge as an oracle ($\hat{\mathcal{D}}_{bias} = \mathcal{D}_{bias}$ and $\hat{\mathcal{D}}_{unbias} = \mathcal{D}_{unbias}$).

4.3. Ablation Study

We conducted an ablation analysis using Corrupted CIFAR-10[1] (bias ratio$= 95\%$), to assess the contribution of each step characterizing our approach.

First, we test the robustness of the classification performance towards the choice of the hyper-parameter γ that governs the amount of data that we assign to the pseudo-labeled subsets (See Figure 3).

Since the dataset is bias, i.e. $|\mathcal{D}_{bias}| >> |\mathcal{D}_{unbias}|$, the choice of a skewed value of γ, i.e. from 65% to 95%, seems reasonable: we observe how the final accuracy does not change sensibly, meaning that the initial training of the network f_ϕ is not a critical step since the biased training samples can be learnt faster than the unbiased ones. In principle it could happen that f_ϕ might not reach the training accuracy of γ, however in all our experiments we were able to reach at least $\gamma = 0.95$.

In Table 4 We also assessed the quality of the splitting obtained, *i.e.*, the influence of the pseudo-labeling (Eq. 2) on the final test accuracy. Considering F1-score (as a measure integrating Precision and Recall) as metric to evaluate the quality of the splitting, we estimate the test accuracies in the ideal case of perfect subdivision between biased and unbiased samples (Oracle, $F1 = 1$), by applying our approach ($F1 = 0.64$), and in case of random split ($F1 = 0.37$). We noted that passing from the oracle conditions (best) to the random split (worst), accuracy drops of 4% in case of our bias identification strategy, and of 10% for the random split, showing a certain robustness to a coarse initial biased/unbiased sample subdivision. In other words, a coarse splitting better than random considerably increase the final performance wrt to ERM training.

Set 1	Set 2	Bias ratio 95% Acc. all	95% Acc. unbias	98% Acc. all	98% Acc. unbias	99% Acc. all	99% Acc. unbias	99.5% Acc. all	99.5% Acc. unbias
No augmentation		58.8%	55.3%	46.1%	41.5%	40.0%	34.8%	33.6%	27.1%
$\hat{\mathcal{D}}_{bias}$	$\hat{\mathcal{D}}_{bias}$	35.2%	29.7%	34.0%	28.4%	32.9%	26.7%	32.0%	27.5%
$\hat{\mathcal{D}}_{unbias}$	$\hat{\mathcal{D}}_{unbias}$	60.2%	63.1%	54.1%	55.3%	48.4%	48.7%	40.4%	42.5%
$\hat{\mathcal{D}}_{bias}$	$\hat{\mathcal{D}}_{unbias}$	**63.8%**	**63.3%**	**56.4%**	**55.9%**	**50.9%**	**49.4%**	**43.1%**	**42.7%**

Table 5. **Ablation analysis on the augmentation strategies.** We report the accuracy resulting from different augmentation strategies and no augmentation, by varying the bias ratio. Our strategy results the winner over all the other mixing policies.

Finally, we tested different strategies to perform data augmentation in the outer loop step. We combined samples from $\hat{\mathcal{D}}_{bias}$ and $\hat{\mathcal{D}}_{unbias}$ (Eq. 4). In Table 4 we report the results when sampling $\hat{x}_1$, $\hat{x}_2$ from different combinations of the subsets. Mixing both samples from $\hat{\mathcal{D}}_{bias}$ overfits the biased data and results in the worst accuracy, while mixing both samples from $\hat{\mathcal{D}}_{unbias}$ increases the generalization over unbiased samples, but provides suboptimal results, especially for the biased subset. Samples from $\hat{\mathcal{D}}_{bias}$ mixed with $\hat{\mathcal{D}}_{unbias}$ corresponds to our policy, which provides the best performance. We also report the baseline case in which no augmentation is performed (first row), i.e. x_{mix}, y_{mix} are just drawn from $\mathcal{D}$, whose results are significantly distant from our proposal. Further ablations details are reported in the Supplementary Material.

5. Conclusions

We proposed a novel solution for the problem of unsupervised debiasing using a meta-learning strategy. After having subdivided by a pseudo-labeling method the training dataset into two subsets of biased and unbiased samples, we treated them as tasks to be learned with a bi-level optimization algorithm. The key idea is the mixing of the two subsets to provide the model with unseen data that can break the spurious correlations between data and class labels.

References

[1] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. pages 0–0, 2018. 1, 3

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020. 1, 3

[3] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. pages 528–539, 2020. 3, 6, 7

[4] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. 31, 2018. 3

[5] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. *CoRR*, abs/1807.04975, 2018. 1

[6] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases, 2019. 1, 2

[7] Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa A. Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement, 2019. 3

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. 70:1126–1135, 06–11 Aug 2017. 2, 3

[9] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *CoRR*, abs/2004.07780, 2020. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 6

[11] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *CoRR*, abs/1903.12261, 2019. 5

[12] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. pages 9012–9020, 2019. 1, 3

[13] Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. *CoRR*, abs/2108.10008, 2021. 3

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015. 6

[15] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. 25:1097–1105, 2012. 6

[17] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 5

[18] Daniel Levy, Yair Carmon, John C. Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization, 2020. 3, 6, 7

[19] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. *CoRR*, abs/1710.03463, 2017. 3

[20] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. pages 9572–9581, 2019. 1, 3, 6, 7

[21] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalisation. 2019. 3

[22] Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. *CoRR*, abs/2107.09044, 2021. 3, 4, 6, 7

[23] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder, 2017. 3

[24] Daniel Moyer, Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. Evading the adversary in invariant representation. *CoRR*, abs/1805.09458, 2018. 3

[25] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. *Advances on Neural Information Processing systems (NeurIPS)*, 2020. 2, 3, 4, 5, 6, 7, 8

[26] Ruggero Ragonesi, Riccardo Volpi, Jacopo Cavazza, and Vittorio Murino. Learning unbiased representations via mutual information backpropagation, 2020. 1, 3

[27] 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. *CoRR*, abs/1911.08731, 2019. 1, 3, 6, 7

[28] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. *CoRR*, abs/2103.02023, 2021. 3

[29] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. *The International Conference on Learning Representations (ICLR)*, 2019. 1

[30] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 6

[31] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning, 2018. 3

[32] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017. 2, 3, 4

[33] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 6