Bias Mimicking: A Simple Sampling Approach for Bias Mitigation

Maan Qraitem[1], Kate Saenko[1,2], Bryan A. Plummer[1]
[1]Boston University [2]MIT-IBM Watson AI Lab
{mqraitem,saenko,bplum}@bu.edu

Abstract

Prior work has shown that Visual Recognition datasets frequently underrepresent bias groups B (e.g. *Female) within class labels Y* (e.g. *Programmers). This dataset bias can lead to models that learn spurious correlations between class labels and bias groups such as age, gender, or race. Most recent methods that address this problem require significant architectural changes or additional loss functions requiring more hyper-parameter tuning. Alternatively, data sampling baselines from the class imbalance literature* (e.g. *Undersampling, Upweighting), which can often be implemented in a single line of code and often have no hyper-parameters, offer a cheaper and more efficient solution. However, these methods suffer from significant shortcomings. For example, Undersampling drops a significant part of the input distribution per epoch while Oversampling repeats samples, causing overfitting. To address these shortcomings, we introduce a new class-conditioned sampling method: Bias Mimicking. The method is based on the observation that if a class c bias distribution,* i.e. $P_D(B|Y=c)$ *is mimicked across every $c' \neq c$, then Y and B are statistically independent. Using this notion, BM, through a novel training procedure, ensures that the model is exposed to the entire distribution per epoch without repeating samples. Consequently, Bias Mimicking improves underrepresented groups' accuracy of sampling methods by 3% over four benchmarks while maintaining and sometimes improving performance over nonsampling methods. Code:* https://github.com/mqraitem/Bias-Mimicking

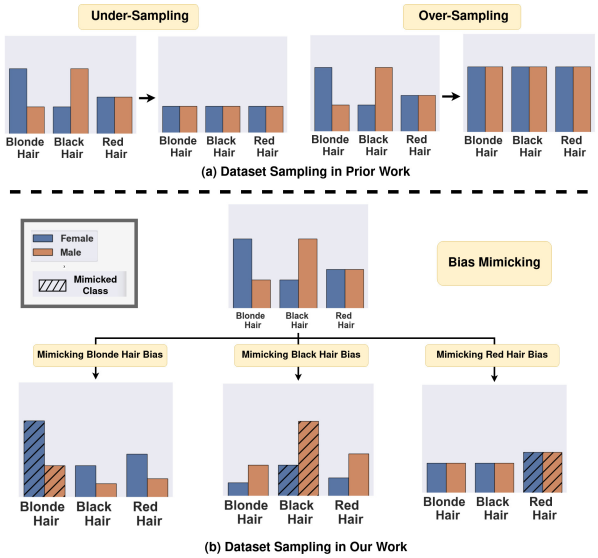


Figure 1. Comparison of sampling approaches for mitigating bias of class labels Y (Hair Color) toward sensitive group labels B (Gender). **(a)** illustrates Undersampling/Oversampling methods that drop/repeat samples respectively from a dataset D per epoch and thus ensure that $P_D(Y|B) = P_D(Y)$. However, dropping samples hurt the model's predictive performance, and repeating samples can cause overfitting with over-parameterized models like neural nets [35]. **(b)** shows our Bias Mimicking approach which subsamples D and produces three variants. Each variant, denoted as $d_c \subset D$, preserves class c samples (*i.e.* mimicked class) and mimics the bias of class c in each $c' \neq c$. This mimicking process, as we show in our work, ensures that $P_{d_c}(Y|B) = P_{d_c}(Y)$. Moreover, by using each d_c separately to train the model, we expose it to all the samples in D per epoch, and since we do not repeat samples in each d_c, our method is less prone to overfitting.

1. Introduction

Spurious predictive correlations have been frequently documented within the Deep Learning literature [34, 38]. These correlations can arise when most samples in class a c (*e.g.* blonde hair) belong to a bias group s (*e.g.* female). Thus, the model might learn to predict classes by using their membership to their bias groups (*e.g.* more likely to predict blonde hair if a sample is female). Mitigating such spurious correlations (Bias) involves decorrelating the model's predictions of input samples from their membership to bias groups. Previous research efforts have primarily focused on model-based solutions. These efforts can be mainly categorized into two directions 1) ensemble-based methods [35], which introduce separate prediction heads for samples from different bias groups 2) methods that introduce additional bias regularizing loss functions and require additional hyper-parameter tuning [12, 15, 26, 27, 33].

Dataset resampling methods, popular within the class imbalance literature [3, 8, 13, 29], present a simpler and cheaper alternative. They do not require hyperparameter tuning or extra model parameters. Therefore, they are faster to train. Moreover, as illustrated in Figure 1(a), they can be extended to Bias Mitigation by considering the imbalance within the dataset subgroups rather than classes. Most common of these methods are Undersampling [3, 13, 29] and Oversampling [35]. They mitigate class imbalance by altering the dataset distribution through dropping/repeating samples, respectively. Another similar solution is Upweighting [4,30], which levels each sample contribution to the loss function by appropriately weighting its loss value. However, these methods suffer from significant shortcomings. For example, Undersampling drops a significant portion of the dataset per epoch, which could harm the models' predictive capacity. Moreover, Upweighting can be unstable when used with stochastic gradient descent [2]. Finally, models trained with Oversampling, as shown by [35], are likely to overfit due to being exposed to repetitive sample copies.

To address these problems, we propose Bias Mimicking (BM): a class-conditioned sampling method that mitigates the shortcomings of prior work. As shown in Figure 1(b), given a dataset D with a set of three classes C, BM subsamples D and produces three different variants. Each variant, $d_c \subset D$ retains every sample from class c while subsampling each $c' \neq c$ such that c' bias distribution, *i.e.* $P_{d_c}(B|Y = c')$, mimics that of c. For example, observe $d_{\text{Blonde Hair}}$ in Figure 1(b) bottom left. Note how the bias distribution of class "Blonde Hair" remains the same while the bias distributions of "Black Hair" and "Red Hair" are subsampled such that they mimic the bias distribution of "Blonde Hair". This mimicking process decorrelates Y from B since Y and B are now statistically independent as we prove in Section 3.1.

The strength of our method lies in the fact that d_c retains class c samples while at the same time ensuring $P_{d_c}(Y|B) = P_{d_c}(Y)$ in each d_c. Using this result, we introduce a novel training procedure that uses each distribution separately to train the model. Consequently, the model is exposed to the entirety of D since each d_c retains class c samples. Refer to Section 3.1 for further details. Note how our method is fundamentally different from Undersampling. While Undersampling also ensures statistical independence on the dataset level, it subsamples every subgroup. Therefore, the training distribution per epoch is a smaller portion of the total dataset D. Moreover, our method is also different from Oversampling since each d_c does not repeat samples. Thus we reduce the risk of overfitting.

In addition to proposing Bias Mimicking, another contribution of our work is providing an extensive analysis of sampling methods for bias mitigation. We found many sampling-based methods were notably missing in the comparisons used in prior work [12,33,35]. Despite their shortcomings, we show that Undersampling and Upweighting are surprisingly competitive on many bias mitigation benchmarks. Therefore, this emphasizes these methods' importance as an inexpensive first choice for mitigating bias. However, in cases where these methods are ineffective, Bias Mimicking bridges the performance gap and achieves comparable performance to nonsampling methods. Finally, we thoroughly analyze our approach's behavior through two experiments. First, we verify the importance of each d_c to the model's predictive performance in Section 4.2. Second, we investigate our method's sensitivity to the mimicking condition in Section 4.3. Both experiments showcase the importance of our design in mitigating bias.

Our contributions can be summarized as:

- We show that simple sampling methods can be competitive on some benchmarks when compared to non sampling state-of-the-art approaches.
- We introduce a novel resampling method: Bias Mimicking that bridges the performance gap between sampling and nonsampling methods; it improves the average underrepresented subgroups accuracy by $> 3\%$ compared to other sampling methods.
- We conduct an extensive empirical analysis of Bias Mimicking that details the method's sensitivity to the Mimicking condition and uncovers insights about its behavior.

2. Related Work

Documenting Spurious Correlations. Bias in Machine Learning can manifest in many ways. Examples include class imbalance [13], historical human biases [32], evaluation bias [8], and more. For a full review, we refer the reader to [23]. In our work, we are interested in model bias that arises from spurious correlations. A spurious correlation results from underrepresenting a certain group of samples (*e.g.* samples with the color red) within a certain class (*e.g.* planes). This leads the model to learn the false relationship between the class and the over-represented group. Prior work has documented several occurrences of this bias. For example, [10,21,31,36] showed that state-of-the-art object recognition models are biased toward backgrounds or textures associated with the object class. [1,7] showed similar spurious correlations in VQA. [38] noted concerning correlations between captions and attributes like skin color in Image-Captioning datasets. Beyond uncovering these correlations within datasets, prior work like [12, 35] introduced synthetic bias datasets where they systematically assessed bias effect on model performance. Most recently, [24] demonstrates how biases toward gender are ubiquitous in the COCO and OpenImages datasets. As the authors demonstrate, these artifacts vary from low-level information (e.g., the mean value of the color channels) to the higher level (e.g., pose and location of people).

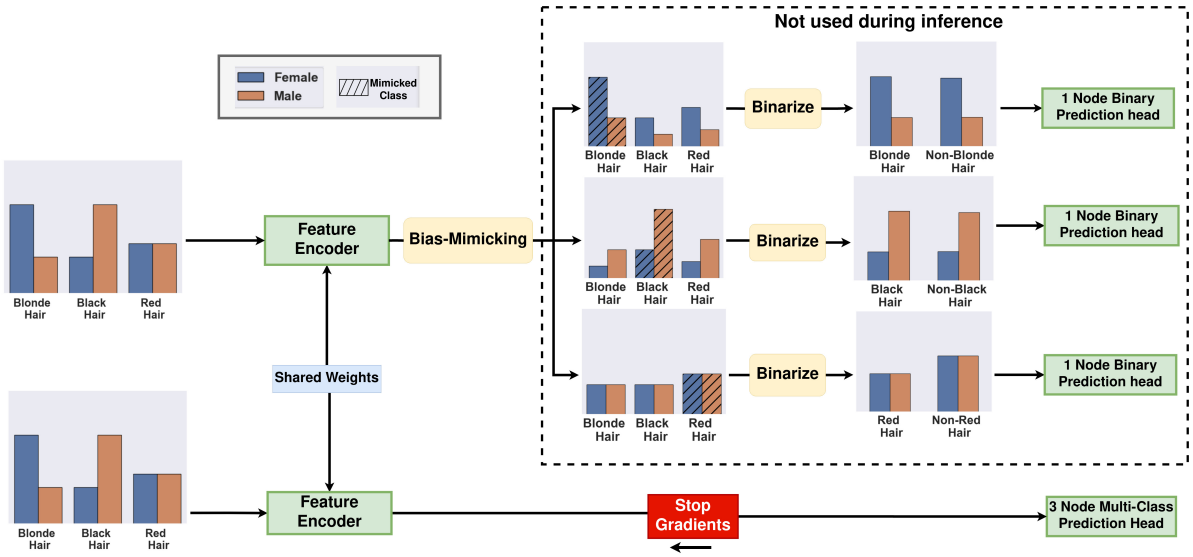


Figure 2. Given a dataset D with three classes, Bias Mimicking subsamples D and produces three different distributions. Each distribution $d_c \subset D$ retains class c samples while subsampling each $c' \neq c$ to ensure the bias distribution of c, *i.e.* $P_{d_c}(B|Y = c)$ is mimicked in each c'. To train an image classification model on the resulting distributions, we binarize each and feed it through a dedicated binary prediction head. Alternatively, we could dedicate a multi-class prediction head for each distribution. However, this alternative will introduce significantly more parameters. Put together, the binary prediction heads are equivalent #parameters wise to a model that uses one multi-class prediction head. To perform inference, we can not use the scores from the binary predictors because they may not be calibrated with respect to each other. Therefore, we train a multi-class prediction head using the debiased feature encoder from Bias Mimicking and freeze the gradient from flowing back to ensure the bias is not relearned.

Nonsampling solutions. In response to the documentation of dataset spurious correlations, several model-focused methods have been proposed [12, 15, 26, 28, 33, 35]. For example, [33] presents a metric learning-based method where model feature space is regularized against learning harmful correlations. [35] surveys several existing methods, such as adversarial training, that randomize the relationship between target classes and bias groups in the feature space. They also present a new approach, domain independence, where different prediction heads are allocated for each subgroup. [28] presents GroupDRO (Distributionally Robust Neural Networks for Group Shifts), a regularization procedure that adapts the model optimization according to the worst-performing group. Most recently, [12] extended contrastive learning frameworks on self-supervised learning [6, 9, 14] to mitigate bias. Our work complements these efforts by introducing a hyperparameter-free sampling algorithm that bridges the performance gap between nonsampling and sampling methods.

Dataset based solutions. In addition to model-based approaches, we can mitigate spurious correlations by fixing the training dataset distribution. Examples include Oversampling minority classes, Undersampling majority ones, and weighting the loss value of different samples to equalize the contribution of dataset subgroups. These approaches are popular within the class imbalance literature [3, 8, 13, 29]. However, as noted in the introduction, some of these methods have been missing in recent visual Bias Mitigation Benchmarks [12, 33, 35]. Thus, we review these methods and describe their shortcomings in Section 3.2. Alternatively, other efforts attempt to fix the dataset distribution by introducing new samples. Examples include [19], where they introduce a new dataset for face recognition balanced among several race groups, and [25], where they used GANs to generate training data that balance the sizes of dataset subgroups. While our work is a dataset-based approach, it differs from these efforts as it does not generate or introduce new samples. Finally, also related to our work are sampling methods like REPAIR [20], where a function is learned to prioritize specific samples and, thus, learn more robust representations.

3. Sampling For Bias Mitigation

In visual bias mitigation, the goal is to train a model that does not rely on spurious signals in the images when making predictions (*e.g.* not using gender signal when predicting hair color). Formally, assume we have a dataset of image/target-classes/bias-group triplets (X, Y, B) where the images X contain components denoted as X_b that determine their bias group memberships B. Furthermore, let C represents the set of possible target-classes, and S represents the set of possible bias groups. We characterise

a model behavior as biased when the model uses the biased image signal, *i.e.* X_b, to predict Y. This behavior might arise when a target class $c \in C$ (*e.g.* blonde hair) is over-represented by images that belong to one bias group $s \in S$ (*e.g.* female) rather than distributed equally among the dataset bias groups. In other words, there exists $s \in B$ for certain class $c \in C$ such that $P_D(B = s|Y = c) >> \frac{1}{|S|}$ where $|S|$ denotes the number of bias groups. For example, if most blonde hair samples were female, the model might use the image components corresponding to gender in making predictions rather than solely relying on hair color.

Our work addresses methods that fix spurious correlations by ensuring statistical independence between Y and B on the dataset level, *i.e.* $P_D(Y|B) = P_D(Y)$. In that spirit, we introduce a new sampling method: Bias Mimicking in Section 3.1. However, as we note in our introduction, some popular sampling solutions in class imbalance literature that could also be applied to Bias Mitigation (*e.g.* Undersampling, Upweighting) are missing from benchmarks used in prior work. Therefore, we briefly review the missing methods in Section 3.2 and describe how Bias Mimicking addresses their shortcomings.

3.1. Bias Mimicking

The goal of our method is to prevent the model from using the biased signal in the images, denoted as X_b, in making predictions $\hat{Y}$. Our approach is inspired by Sampling methods, which mitigate this problem by enforcing statistical Independence between the target labels Y and the bias groups B in the dataset, *i.e.*, $P_D(Y|B) = P_D(Y)$. However, simple sampling methods like Oversampling, Undersampling, and Upweighting need to be improved as we discuss in the introduction. We address these shortcomings in our proposed sampling algorithm: Bias Mimicking.

Mimicking Distributions The key to our algorithm is the observation that if the distribution of bias groups with respect to a class c; *i.e.*, $P_D(B|Y = c)$, was the same across every $c \in C$, then B is statistically independent from Y. For example, consider each resulting distribution from our Bias Mimicking process in Figure 2. Note how in each resulting distribution, the distribution of bias group "Gender" is the same across every class "Hair Color" which ensures statistical independence between "Gender" and "Hair Color" on the dataset level. Formally:

Proposition 1. *Given dataset D, target classes set C, bias groups set S, target labels Y, bias groups labels B, and bias group $s \in S$ if $P_D(B = s|Y = i) = P_D(B = s|Y = j) \quad \forall i, j \in C$, then $P_D(Y = c|B = s) = P_D(Y = c) \quad \forall c \in C$.*

Note that ensuring this proposition holds for every $s \in S$ (*i.e.* the distribution of $P_D(B|Y = c)$ is the same for each $c \in C$) implies that $P_D(Y|B) = P_D(Y)$. Proving the proposition above is a simple application of the law of total probability. Given $s \in S$, then

$$P_D(B = s) = \sum_{c \in C} P_D(B = s|Y = c)P_D(Y = c) \quad (1)$$

Given our assumption of bias mimicking, *i.e.* $P_D(B = s|Y = i) = P_D(B = s|Y = j) \quad \forall i, j \in C$, then we can rewrite (1) $\forall c \in C$ as:

$$P_D(B = s) = P_D(B = s|Y = c) \sum_{c' \in C} P_D(Y = c') \quad (2)$$

$$= P_D(B = s|Y = c) \quad (3)$$

From here, using Bayesian probability and the result from (3), we can write, $\forall c \in C$:

$$P_D(Y = c|B = s) = \frac{P_D(B = s|Y = c)P_D(Y = c)}{P_D(B = s)} \quad (4)$$

$$= \frac{P_D(B = s)P_D(Y = c)}{P_D(B = s)} \quad (5)$$

$$= P_D(Y = c) \quad (6)$$

We use this result to motivate a novel way of subsampling the input distribution that ensures the model is exposed to every sample in the dataset. To that end, for every class $c \in C$, we produce a subsampled version of the dataset D denoted as $d_c \subset D$. Each d_c preserves its respective class c samples, *i.e.*, all the samples from class c remain in d_c, while subsampling every $c' \neq c$ such that the bias distribution in class c; *i.e.* $P_{d_c}(B|Y = c)$, is mimicked in each c'. Formally:

$$P_{d_c}(B = s|Y = c) = P_{d_c}(B = s|Y = c') \quad \forall s \in S. \quad (7)$$

Indeed, as Figure 2 illustrates, a dataset of three classes will be subsampled in three different ways. In each version, a class bias is mimicked across the other classes. Note that there is not unique solution for mimicking distributions. However, we can constrain the mimicking process by ensuring that we retain the most number of samples in each subsampled $c' \neq c$. We enforce this constraint naturally through a linear program. Denote the number of samples that belong to class c' as well as the bias group s as $l_s^{c'}$. Then we seek to optimize each $l_s^{c'}$ as follows:

$$\text{(P.1)} \quad \begin{aligned} \max \quad & \sum_s l_s^{c'} \\ \text{s.t.} \quad & l_s^{c'} \leq |D_{c',s}| && s \in S \quad (8) \\ & \frac{l_s^{c'}}{\sum_s l_s^{c'}} = P_D(B = s|Y = c) && s \in S \quad (9) \end{aligned}$$

where $|D_{c,s}|$ represents the number of samples that belong to both class c and bias group s. This linear program returns

the distribution with most number of retained samples while ensuring that the mimicking condition holds.

Training with Bias Mimicking We use every resulting $d_c \subset D$ to train the model. However, we train our model such that it sees each d_c through a different prediction head to ensure that the loss function does not take in gradients of repetitive copies, thus risking overfitting. A naive way of achieving this is to dedicate a multi-class prediction head for each d_c. However, this choice introduces $(|C|-1)$ additional prediction heads compared to a model that uses only one. To avoid this, we binarise each d_c and then use it to train a one-vs-all binary classifier BP_c for each c. Each head is trained on image-target pairs from its corresponding distribution d_c as Figure 2 demonstrates. Note that each binary predictor is simply one output node. Therefore, the binary predictors introduce no extra parameters compared to a multi-class head. Finally, given our setup, a prediction head might see only a small number of samples at certain iterations. We mitigate this problem by accumulating the gradients of each prediction head across iterations and only backpropagating them once the classifier batch is full.

Inference with Bias Mimicking Using the binary classifiers during inference is challenging since each was trained on different distributions, and their scores, therefore, may not be calibrated. However, Bias Mimicking minimizes the correlation between Y and B within the feature space. Thus, we exploit this fact by training a multi-class prediction head over the feature space using the original dataset distribution. However, this distribution is biased; therefore, we stop gradients from flowing into the model backbone when training that layer to ensure that the bias is not learned again in the feature space, as Figure 2 demonstrates.

Cost Analysis Unlike prior work [12, 33], bias mimicking involves no extra hyper-parameters. The debiasing is automatic by definition. This means that the hyper-parameter search space of our model is smaller than other bias mitigation methods like [12, 28]. Consequently, our method is cheaper and more likely to generalize to more datasets as we show in Section 4.1. Moreover, note that the additional input distributions d_c do not result in longer epochs; we make one pass only over each sample x during an epoch and apply its contribution to the relevant BP_y(s). Our only additional cost is solving the linear program P.1 and training the multi-class prediction head. However, training the linear layer is simple, fast, and cheap since it is only *one* linear layer. Moreover, the linear program is solved through modern and efficient implementations that take seconds and only needs to be done once for each dataset.

3.2. Simple Sampling Methods

As discussed in the introduction, the class imbalance literature is rich with dataset sampling solutions [3,5,8,13,29]. These solutions address class imbalance by balancing the distribution of classes. Thus, they can be extended to Bias Mitigation by balancing groupings of classes and bias groups (*e.g.* group male-black hair). Prior work in Visual Bias Mitigation has explored one of these solutions: Oversampling [35]. However, other methods like Undersampling [3, 13, 29] and Upweighting [4, 30] are popular alternatives to Oversampling. Both methods, however, have not been benchmarked in recent Visual Bias Mitigation work [12, 33]. We review these sampling solutions below and note how Bias Mimicking addresses their shortcomings.

Undersampling drops samples from the majority classes until all classes are balanced. We can extend this solution to Bias Mitigation by dropping samples to balance dataset subgroups where a subgroup includes every sample that shares a class c and bias group s, *i.e.*, $D_{c,s} = \{(x, y, s) \in D \text{ s.t } Y = c, B = s\}$. Then, we drop samples from each subgroup until each has a size equal to $\min_{c,s} |D_{c,s}|$. Thus, in each epoch, the number of samples the model can see is limited by the size of the smallest subgroup. While we can mitigate this problem by resampling the distribution each epoch, the model, nevertheless, has to be exposed to repeated copies of the minority subgroup every time it sees new samples from the majority subgroups, which may compromise the predictive capacity of our model, as our experimental results demonstrate. *Bias Mimcking* addresses this shortcoming by using each $d_c \subset D$, thus ensuring that the model is exposed to all of D every epoch.

Oversampling solves class imbalance by repeating copies of samples to balance the number of samples in each class. Similarly to Undersampling, it can also be easily extended to Bias Mitigation by balancing samples across sensitive subgroups. In our implementation, we determine the maximum size subgroup (*i.e.* $\max_{c,s} |D_{c,s}|$ where $D_{c,s} = \{(x, y, s) \in D \text{ s.t } Y = c, B = s\}$) and then repeat samples accordingly in every other subgroup. However, repeating samples as [35] shows, cause overfitting with overparameterized models like neural nets. *Bias Mimicking addresses* this problem by not repeating samples in each subsampled version of the dataset $d_c \subset D$.

Upweighting Upweighting levels the contribution of different samples to the loss function by multiplying its loss value by the inverse of the sample's class frequency. We can extend this process to Bias Mitigation by simply considering subgroups instead of classes. More concretely, assume model weights w, sample x, class c, bias s, and subgroup $D_{c,s} = \{(x, y, s) \in D \text{ s.t } Y = c, B = s\}$, then the model optimizes:

$$L = E_{x,c,s}\Big[\frac{1}{p_D(x \in D_{c,s})} l(x, c; w)\Big]$$

A key shortcoming of Upweighting is its instability when used with stochastic gradient descent [2]. Indeed,

			Non Sampling Methods				Sampling Methods			
		Vanilla	Adv [11]	G-DRO [27]	DI [35]	BC+BB [12]	OS [35]	UW [4]	US [13]	BM (ours)
UTK-Face	UA	72.8±0.2	70.2±0.1	74.2±0.9	75.5±1.1	78.9±0.5	76.6±0.3	78.8±1.2	78.2±1.0	79.7±0.4
Age	BC	47.1±0.3	44.1±1.2	75.9±2.9	58.8±3.0	71.4±2.9	58.1±1.2	77.2±3.8	69.8±6.8	79.1±2.3
UTK-Face	UA	88.4±0.2	86.1±0.5	90.8±0.3	90.7±0.1	91.4±0.2	91.3±0.6	89.7±0.6	90.8±0.2	90.8±0.2
Race	BC	80.8±0.3	77.1±1.3	90.2±0.3	90.9±0.4	90.6±0.5	90.0±0.7	89.2±0.4	89.3±0.6	90.7±0.5
CelebA	UA	82.4±1.3	82.4±1.3	90.4±0.4	90.9±0.5	90.4±0.2	88.1±0.5	91.6±0.3	91.1±0.2	90.8±0.4
Blonde	BC	66.3±2.8	66.3±2.8	89.4±0.5	86.1±0.8	86.5±0.5	80.1±1.2	88.3±0.5	88.5±1.8	87.1±0.6
CIFAR-S	UA	88.7±0.1	81.8±2.5	89.1±0.2	92.1±0.0	90.9±0.2	87.8±0.2	86.5±0.5	88.2±0.4	91.6±0.1
	BC	82.8±0.1	72.0±0.2	88.0±0.2	91.9±0.2	89.5±0.7	82.5±0.3	80.0±0.7	83.7±1.2	91.1±0.1
Average	UA	83.0±0.4	80.1±1.1	86.1±0.4	87.3±0.4	**87.9±0.3**	85.9±0.4	86.6±0.6	87.0±0.4	**88.2±0.3**
	BC	69.2±0.8	64.8±1.4	**85.8±1.0**	81.9±1.1	84.5±1.1	77.6±0.8	83.6±1.3	82.8±2.6	**87.0±0.9**

Table 1. **Results** compare methods Adversarial training (Adv) [16], GroupDRO (G-DRO) [27], Domain Independence DI [35], Bias Contrastive and Bias-Contrastive and Bias-Balanced Learning (BC+BB) [12], Undersampling (US) [13], Upweighting (UW) [4], and Bias Mimicking (BM, ours), on the CelebA, UTK-face, and CIFAR-S dataset. Methods are evaluated using Unbiased Accuracy [12] (UA), Bias-conflict [12] (BC). Given the methods' grouping: Sampling/Non Sampling, the Underlined numbers indicate the best method per group on each dataset while **bolded numbers** indicate the best method per group on average. See Section 4.1 for discussion.

we demonstrate this problem in our experiments in Section 4 where Upweighting does not work well on some datasets. *Bias Mimicking addresses* this problem by not using weights to scale the loss function.

4. Experiments

We report our method performance on four total benchmarks: three binary and one multi-class classification benchmark in Section 4.1. In addition to reporting our method results, we include vanilla sampling methods, namely Undersampling, Upweighting, and Oversampling, which, as noted in our introduction, have been missing from recent Bias Mitigation work [12, 28, 35]. Furthermore, we report the averaged performance overall benchmarks to highlight methods that generalize well to all datasets. We then follow up our results with two main experiments that analyze our method's behavior. The first experiment in Section 4.2 analyzes the contribution of each subsampled version of the dataset: d_c to model performance. The second experiment in Section 4.3 is a sensitivity analysis of our method to the mimicking condition.

4.1. Main Results

Datasets We report performance on three main datasets: CelebA dataset [22] UTKFace dataset [37] and CIFAR-S benchmark [35]. Following prior work [12, 33], we train a binary classification model using CelebA where BlondHair is a target attribute and Gender is a bias attribute. We use [12] split where they amplify the bias of Gender. Note that prior work [12, 33] used the HeavyMakeUp attribute in their CelebA benchmark. However, during our experiments, we found serious problems with the benchmark. Refer to the supplementary for model details. Therefore, we skip this benchmark in our work. For UTKFace, we follow [12] and do the binary classification with Race/Age as the sensitive attribute and Gender as the target attribute. We use [12] split for both tasks. With regard to CIFAR-S, The benchmark synthetically introduces bias into the CIFAR-10 dataset [18] by converting a subsample of images from each target class to a gray-scale. We use [35] version where half of the classes are biased toward color and the other half is biased toward gray where the dominant sensitive group in each target represents 95% of the samples.

Metrics Using Average accuracy on each sample is a misleading metric. This is because the test set could be biased toward some subgroups more than others. Therefore, the metric does not reflect how the model performs on *all* subgroups. Methods, therefore, are evaluated in terms of Unbiased Accuracy [12], which computes the accuracy of each subgroup separately and then returns the mean of accuracies, and Bias-Conflict [12], which measures the accuracy on the minority subgroups.

Baselines We report several baselines from prior work. First, we report "Bias-Contrastive and Bias-Balanced Learning" (BC + BB) [12], which uses a contrastive learning framework to mitigate bias. Second, domain-independent (DI) [35] uses an additional prediction head for each bias group. GroupDRO [27] which optimizes a worst-group training loss combined with group-balanced

	UA_{c_1}	UA_{c_2}	UA
(d_{c_1})	82.5	74.2	78.4±0.5
(d_{c_2})	73.1	67.9	70.5±2.3
(d_{c_1}, d_{c_2})	<u>84.4</u>	<u>75.3</u>	**79.8±0.9**

(a) Utk-Face/Age

	UA_{c_1}	UA_{c_2}	UA
(d_{c_1})	<u>90.7</u>	85.1	87.9±0.5
(d_{c_2})	84.8	<u>91.5</u>	88.1±0.6
(d_{c_1}, d_{c_2})	90.5	91.1	**90.8±0.2**

(b) Utk-Face/Race

	UA_{c_1}	UA_{c_2}	UA
(d_{c_1})	<u>91.5</u>	90.3	**90.9±0.1**
(d_{c_2})	82.2	<u>96.5</u>	89.3±0.1
(d_{c_1}, d_{c_2})	88.1	94.2	**90.8±0.4**

(c) CelebaA/Blonde

Table 2. We investigate the effect of each resampled version of the dataset d_c on model performance using the binary classification tasks outlined in Section 4.1. Thus, Bias Mimicking results in two subsampled distributions d_{c_1} and d_{c_2}. We then train three different models: 1) a model trained on only d_{c_1}, 2) a model trained on only d_{c_2} and a model trained on both (default choice for Bias Mimicking). We use the Unbiased Accuracy metric (UA). Furthermore, we report UA on class 1: UA_1 and class 2: UA_2 separately by averaging the accuracy over the class's relevant subgroups. Refer to section 4.2 for discussion.

resampling. Adversarial Learning (Adv) w/ uniform confusion [11] introduces an adversarial loss that seeks to randomize the model's feature representation of Y and B. Furthermore, we expand the benchmark by reporting the performance of sampling methods outlined in Section 3. Note that since Undersampling and Oversampling result in smaller/larger distributions per epoch than other methods, we adjust their number of training epochs such that the model sees the same number of data batches throughout the training process as other methods in our experiment. Refer to the Supplementary for further details. Finally, all reported methods are compared to a "vanilla" model trained on the original dataset distribution with a Cross-Entropy loss. All baselines are trained using the same encoder. Refer to the Supplementary for further details.

Results in Table 1 demonstrates how BM is the most robust sampling method. It achieves strong performance on every dataset, unlike other sampling methods. For example, Undersampling while performing well on CelebA (likely because the task of predicting hair color is easy), performs considerably worse on every other dataset. Moreover, Oversampling, performs consistently worse than our method. This aligns with [35] observation that neural nets tend to overfit when trained on oversampled distributions. Finally, while Upweighting maintains strong performance on CelebA and UTK-Face, it falls behind on CIFAR-S. This is because the method struggled to optimize, a property of Upweighting that has been noted and discussed in [2]. BM, consequently, improves over Upweighting on CIFAR-S by over $\%10$ on the Bias Conflict metric.

While our method is the only sampling method that shows consistently strong performance, vanilla sampling methods are surprisingly effective on some datasets. For example, Upweighting is effective on UTK-Face and CelebA, especially when compared to non-sampling methods. Moreover, Undersampling similarly is more effective than non-sampling methods on CelebA. This is evidence that when enough data is available and the task is "easy" enough, then simple Undersampling is effective. These results are important because they show how simply fixing the dataset distribution through sampling methods could be a strong baseline. We encourage future work, therefore, to add these sampling methods as part of their baselines.

With respect to non-sampling methods, our method (BM) maintains strong comparable performance. This strong performance comes with ***no*** additional loss functions or any model modification which limits the hyper-parameter space and implementation complexity of our method compared to non-sampling methods. For example, BC+BB [12] requires a scalar parameter to optimize an additional loss. It also requires choosing a set of augmentation functions for its contrastive learning paradigm, which must be tuned according to the dataset. GroupDRO [28], on the other hand, requires careful L2 regularization as well as tuning their "group adjustment" hyper parameter. As a result, BM is faster and cheaper to train.

4.2. The Effect of Each d_c on Model Performance

For each class, $c \in C$, Bias Mimicking returns a subsampled version of the dataset $d_c \subset D$ where class c samples are preserved, and the bias of c is mimicked in the other classes. In this section, we investigate the effect of each d_c on performance. We use the binary classification tasks in Section 4.1. For each task, thus, we have two versions of the dataset d_{c_1}, d_{c_2} where c_1 is the first class and c_2 is the second. We compare three different models performances: model (1) trained on only d_{c_1}, model (2) trained on only d_{c_2}, and finally, model (3) trained on both (d_{c_1}, d_{c_2}). The last version is the one used by our method. We use the Unbiased Accuracy Metric (UA). We also break down the metric into two versions: UA_1 where accuracy is averaged over class 1 subgroups, and UA_2 where accuracy is averaged over class 2 subgroups. Observe the results in Table 2. Overall, the model trained on (d_{c_1}) performs better at predicting c_1 but worse at predicting c_2. We note the same trend with the model trained on (d_{c_2}) but in reverse. This disparity in performance harms the average Unbiased Accuracy (UA). However, a model trained on (d_{c_1}, d_{c_2}) balances both accuracies and achieves the best total Unbiased Accuracy (UA). These results emphasize the importance of each d_c for good performance.

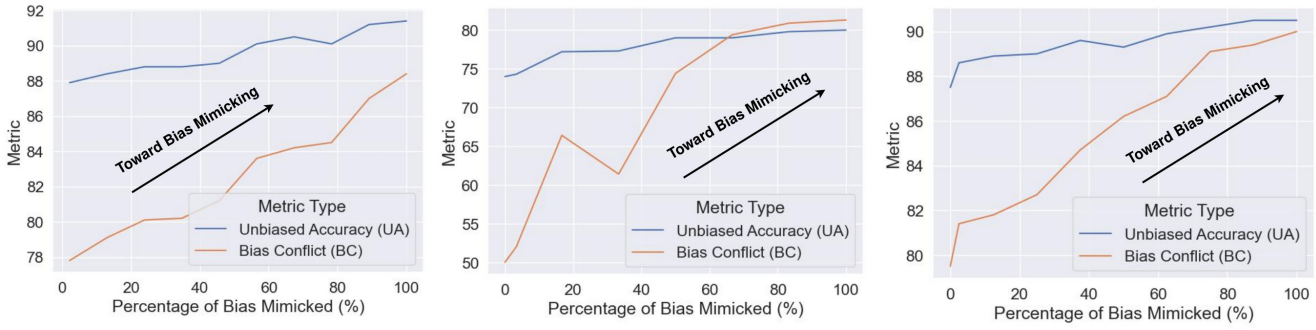


Figure 3. **Model Sensitivity to Bias Mimicking** We test our method's sensitivity to the Bias Mimicking condition in Equation 7. To that end, we simulate multiple scenarios where we mimic the bias by $x\% \in \{0, 100\}$ (x-axis) such that 0% represents no modification to the distribution and 100% represents complete Bias Mimicking. We report results on the three binary classification benchmarks defined in Section 4.1. Refer to Section 4.3 for further details and discussion.

4.3. Model Sensitivity to the Mimicking Condition

In this section, we test the model sensitivity to the mimicking condition outlined in Eq. 7. To that end, we consider the binary classification benchmarks in Section 4.1. These benchmarks have two classes, c_1 and c_2, and two bias groups, b_1 and b_2. Each class is biased toward one of the bias groups. We vary the mimicking process according to a percentage value where for value 0%, the resulting d_c(s) are the same as the original training distribution, and for value $x > 0\%$, $d_c(s)$ are subsampled from the main training distribution such that the bias is mimicked by $x\%$. Note how, therefore, 100% represents the full mimicking process as outlined in Eq. 7 whereas $x = 50\%$ mimicks the bias by half. Formally, if c_2 needs to mimick c_1 and c_1 is biased toward b_1 then

$$P_{d_{c_1}}(B = b_1|Y = c_2) = \frac{1}{2} P_{d_{c_1}}(B = b_1|Y = c_1)$$

Observe the result in Figure 3. Note that as the percentage of bias Mimicked decreases, the BC and UA decrease as expected. This is because $P_D(Y|B) \neq P_D(Y)$ following proposition 1. More interestingly, note how the Bias Conflict (the accuracy over the minority subgroups) decreases much faster. The best performance is achieved when $x\%$ is indeed 100% and thus $P_D(Y|B) = P_D(Y)$. From this analysis, we conclude that the Bias Mimicking condition is critical for good performance.

5. Conclusion

In this paper, we observed that hyper-parameter-free sampling methods for bias mitigation like Undersampling and Upweighting were missing from recent benchmarks. Therefore, we benchmarked these methods and concluded that some are surprisingly effective on many datasets. However, on some others, their performance significantly lagged behind non sampling methods. Motivated by this observation, we introduced a novel sampling method: Bias Mimicking. The method retained the simplicity of sampling methods while bridging the performance gap between sampling and non sampling methods. Furthermore, we extensively analyzed the behavior of Bias Mimicking, which emphasized the importance of our design. We hope our new method can reestablish sampling methods as a promising direction for bias mitigation.

Limitations And Future Work We demonstrated that dataset re-sampling methods are simple and effective tools in mitigating bias. However, we recognize that the explored bias scenarios in prior and our work are still limited. For example, current studies only consider mutually exclusive sensitive groups. Thus, a sample can belong to only one sensitive group at a time. How does relaxing this assumption, *i.e.* intersectionality, impact bias? Finally, the dataset re-sampling methods presented in this work are effective at mitigating bias when enough samples from all dataset subgroups are available. However, it is unclear how these methods could handle bias scenarios when one class is completely biased by one sensitive group. Moreover, sampling methods require full knowledge of the bias groups labels at training time. Collecting annotations for these groups could be expensive. Future work could benefit from building more robust models independently of bias groups labels.

Acknowledgements This material is based upon work supported, in part, by DARPA under agreement number HR00112020054 and the National Science Foundation, including under Grant No. DBI-2134696. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the supporting agencies.

References

[1] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4971–4980, 2018. 2

[2] Jing An, Lexing Ying, and Yuhua Zhu. Why resampling outperforms reweighting for correcting sampling bias with stochastic gradients. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2, 5, 7

[3] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. 2, 3, 5

[4] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 872–881. PMLR, 09–15 Jun 2019. 2, 5, 6, 11

[5] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 06 2002. 5

[6] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 3

[7] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. 2

[8] Joy Buolamwini et al. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *ACM FAccT*, 2018. 2, 3, 5

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 3

[10] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 2

[11] Judy Hoffman, Eric Tzeng, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4068–4076, 2015. 6, 7, 11

[12] Youngkyu Hong and Eunho Yang. Unbiased classification through bias-contrastive and bias-balanced learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 1, 2, 3, 5, 6, 7, 10, 11, 12

[13] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, pages 429–449, 2002. 2, 3, 5, 6, 11, 12

[14] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020. 3

[15] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9004–9012, 2019. 1, 3

[16] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6

[17] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 12

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012. 6

[19] Kimmo Kärkkäinen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1547–1557, 2021. 3

[20] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9564–9573, 2019. 3

[21] Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan Yuille, and Cihang Xie. Shape-texture debiased neural network training. *arXiv preprint arXiv:2010.05981*, 2020. 2

[22] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 6, 10, 11, 12

[23] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021. 2

[24] Nicole Meister, Dora Zhao, Angelina Wang, Vikram V. Ramaswamy, Ruth C. Fong, and Olga Russakovsky. Gender artifacts in visual datasets. *ArXiv*, abs/2206.09191, 2022. 2

[25] Vikram V. Ramaswamy, Sunnie S. Y. Kim, and Olga Russakovsky. Fair attribute classification through latent space de-biasing. *CoRR*, abs/2012.01469, 2020. 3

[26] Hee Jung Ryu, Margaret Mitchell, and Hartwig Adam. Improving smiling detection with race and gender diversity. *CoRR*, abs/1712.00193, 2017. 1, 3

[27] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *ICLR*, 2020. 1, 6, 11, 12

[28] 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. In *International Conference on Learning Representations (ICLR)*, 2020. 3, 5, 6, 7

[29] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8346–8356. PMLR, 13–18 Jul 2020. 2, 3, 5

[30] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 10 2000. 2, 5

[31] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: Learning to overcome contextual bias. In *CVPR*, 2020. 2

[32] Harini Suresh and John V. Guttag. A framework for understanding unintended consequences of machine learning. *ArXiv*, abs/1901.10002, 2019. 2

[33] Enzo Tartaglione, Carlo Alberto Barbano, and Marco Grangetto. End: Entangling and disentangling deep representations for bias correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13508–13517, June 2021. 1, 2, 3, 5, 6, 11

[34] Angelina Wang, Arvind Narayanan, and Olga Russakovsky. Vibe: A tool for measuring and mitigating bias in image datasets. *CoRR*, abs/2004.07999, 2020. 1

[35] Zeyu Wang, Klint Qinami, Ioannis Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 5, 6, 7, 11, 12

[36] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *ArXiv preprint arXiv:2006.09994*, 2020. 2

[37] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *CVPR*, 2017. 6, 12

[38] Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2

A. Sampling methods Impact on Multi-Class Classification Head

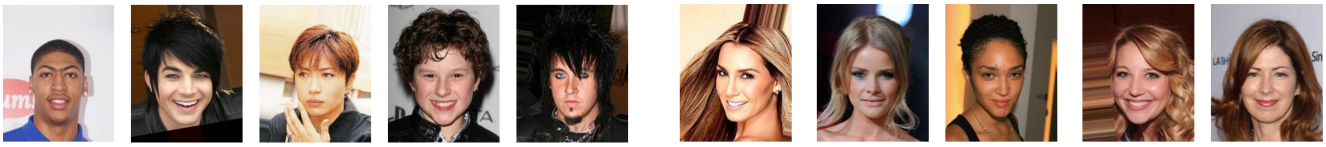
Bias Mimicking produces a binary version d_c of the dataset D for each class c. Each d_c preserves class c samples while undersampling each c' such that the bias within c' mimics that of c. A debiased feature representation is then learned by training a binary classifier for each d_c. When the training is done, using the scores from each binary predictor for inference is challenging. This is because each predictor is trained on a different distribution of the data, so the predictors are uncalibrated with respect to each other. Therefore, to perform inference, we train a multi-class prediction head using the learned feature representations and the original dataset distribution. Moreover, we prevent the gradients from flowing into the feature space since the original distribution is biased. Note that we rely on the assumption that the correlation between the target labels and bias labels are minimized in the feature space, and thus the linear layer is unlikely to relearn the bias. During our experiments outlined in Section 4, we note that this approach was sufficient to obtain competitive results. This section explores whether we can improve performance by using sampling methods to train the linear layer. To that end, observe results in Table 3. We underline the rows where the sampling methods make improvements. We note that the sampling methods did not improve performance for three of the four benchmarks in our experiments. However, on CelebA, we note that the sampling methods marginally improved performance. We suspect this is because a small amount of the bias might be relearned when training the multi-class prediction head since the input distribution remains biased.

		BM	BM+US	BM+UW	BM+OS
UTK-Face	UA	79.7±0.4	79.2±1.0	79.9±0.2	79.3±0.2
Age	BC	79.1±2.3	77.6±0.9	77.5±1.7	78.7±1.6
UTK-Face	UA	90.8±0.2	90.9±0.5	91.1±0.2	90.7±0.4
Race	BC	90.7±0.5	91.1±0.3	91.6±0.1	90.9±0.5
CelebA	UA	90.8±0.4	91.1±0.2	91.1±0.4	91.1±0.1
Blonde	BC	87.1±0.6	<u>87.9±0.3</u>	<u>87.9±0.7</u>	<u>87.7±0.4</u>
CIFAR-S	UA	91.6±0.1	91.7±0.1	91.8±0.0	91.6±0.2
	BC	91.1±0.1	91.2±0.2	91.4±0.2	91.2±0.2

Table 3. **Sampling for multi-class prediction head** compare the effects of using different sampling methods to train the multi-class prediction in our proposed method: Bias Mimicking. We underline results where sampling methods make significant improvements. Refer to Section A for discussion.

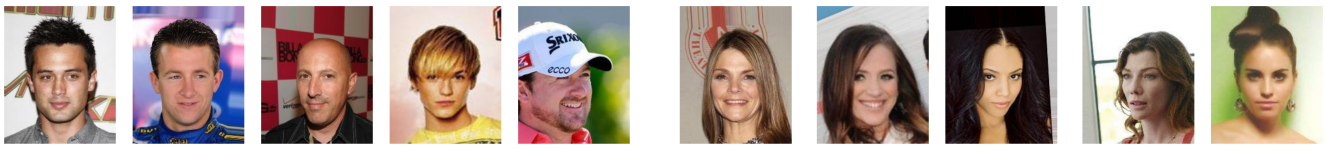
B. Heavy Makeup Benchmark

Prior work [12] uses the Heavy Makeup binary attribute prediction task from CelebA [22] as a benchmark for bias mitigation, where Gender is the sensitive attribute. In this experiment, Heavy Makeup's attribute is biased toward the sensitive group: Female. We note that the notion of "Heavy Makeup" is quite subjective. The attribute labels may vary significantly according to cultural elements, lighting conditions, and camera pose considerations. Thus, we expect a



(a) Male - Heavy Makeup

(a) Female - Heavy Makeup



(b) Male - Non Heavy Makeup

(b) Female - Non Heavy Makeup

Figure 4. Randomly sampled images from the four subgroups: Female-Heavy Makeup, Female-Non-Heavy Makeup, Male-Heavy Makeup, and Male-Non-Heavy Makeup in CelebA dataset [22]. Note the that there is not a clearly differentiating signal for the attribute Heavy Makeup. Refer to Section B for discussion.

		Nonsampling Methods					Sampling Methods				
		Vanilla	Adv [11]	G-DRO [27]	DI [35]	BC+BB [12]	OS [35]	UW [4]	US [13]	BM	BM + OS
CelebA	UA	90.9±0.1	90.7±0.2	93.2±0.1	92.0±0.1	92.4±0.1	92.4±0.3	92.8±0.0	92.6±0.1	92.7±0.1	92.7±0.1
Smiling	BC	84.3±0.2	84.7±0.4	92.2±0.1	91.3±0.2	92.6±0.1	91.5±0.2	92.4±0.2	92.1±0.2	92.3±0.2	92.2±0.1
CelebA	UA	86.3±0.7	87.1±0.3	88.5±0.2	86.7±0.7	87.7±0.1	87.6±0.3	88.5±0.1	88.4±0.2	87.6±0.7	88.5±0.1
Black Hair	BC	82.7±0.6	83.4±0.5	88.3±0.4	86.6±1.2	86.6±0.3	85.6±0.6	88.0±0.2	87.3±0.1	87.8±1.3	88.5±0.7
Average	UA	88.6±0.4	88.9±0.2	**90.8±0.1**	89.3±0.4	90.0±0.1	90.0±0.3	**90.6±0.1**	**90.5±0.1**	**90.2±0.4**	**90.6±0.1**
	BC	83.5±0.4	84.0±0.4	**90.2±0.2**	88.9±0.7	89.6±0.2	88.5±0.4	**90.2±0.2**	89.7±0.1	**90.0±0.7**	**90.3±0.4**

Table 4. Expanded benchmarks from the CelebA dataset [22]. Refer to Section C for discussion.

fair amount of label noise, *i.e.*, inconsistency with label assignment. We document this problem in a Quantitative and Qualitative analysis below.

Quantitative Analysis: We randomly select a total of 200 pairs of positive and negative images. We ensure the samples are balanced among the four possible pairings, *i.e.*, (Heavy Makeup-Male, Non Heavy Makeup-Female), (Heavy Makeup-Female, Non Heavy Makeup-Male), (Heavy Makeup-Male, Non Heavy Makeup-Male), (Heavy Makeup-Female, Non Heavy Makeup-Female). We asked three independent annotators to label which image in the pair is wearing "Heavy Makeup". Then, we calculate the percentage of disagreement between the three annotators and the ground truth labels in the dataset. We note that $32.3\% \pm 0.02$ of the time, the annotators on average disagreed with the ground truth. The noise is further amplified when the test set used in [12] is examined. In particular, Male-Heavy Make up (an under-represented subgroup) only contains 9 testing samples. We could not visually determine whether 4 out of these 9 images fall under Heavy Makeup. Out of the 5 remaining images, 3 are of the same person from different angles. Thus, given the noise in the training set, the small size of the under-represented group in the test set, and its label noise, we conclude that results from this benchmark will not be reliable and exclude it from our experiments.

Qualitative Analysis: We sample random 5 images from the following subgroups: Female-Heavy Makeup, Female-Non-Heavy Makeup, Male-Heavy Makeup, and Male-Non-Heavy Makeup (Fig 4). It is clear from the Figure that there is no firm agreement about the definition of Heavy Makeup.

C. Additional Benchmarks

In Section B, we note that the CelebA attribute Heavy-Makeup usually used in assessing model bias in prior work [12, 33] is a noisy attribute, *i.e.*, labels are inconsistent. Therefore, we choose not to use it in our experiments. Alternatively, we provide results on additional attributes where labels are more likely to be consistent. To that end, we choose to classify the attributes: Smiling and Black Hair, where Gender is the bias variable. The original distribution of each attribute is not sufficiently biased with respect to Gender to note any significant change in performance. Thus, we subsample each distribution to ensure that each

	Learning Rate	Weight Reg	Group Adjustment
UTK-Face Age	0.001	0.01	4
UTK-Face Race	0.001	0.001	4
CelebA Blonde	0.001	0.1	3
CelebA Smiling	0.0001	0.01	2
CelebA Black Hair	0.0001	0.01	3
CIFAR-S	0.01	0.01	5

Table 5. Hyperparameters used for GroupDRO [27]. Refer to Section D for further discussion.

attribute is biased toward Gender. We provide the splits for the resulting distributions in the attached code base. Refer for Table 4 for results.

Note that our method Bias Mimicking performance marginally lags behind other methods when predicting "Black Hair" attribute. However, when the multi-class prediction layer is trained with an oversampled distribution (BM + OS), then the gap is bridged. This is consistent with the observation in Table 3 where oversampling marginally improves our method performance on CelebA. These observations indicate that on some benchmarks, a small amount of the bias might be relearned through the multi-class prediction head. To ensure that this bias is mitigated, it is sufficient to oversample the input distribution. Moreover, since oversampling the input distribution does not change performance on other datasets as indicated in Table 3, we recommend that the input distribution for the multi-class prediction head is oversampled to ensure the best performance.

Overall, (BM + OS) performs comparably to sampling and nonsampling methods. This is consistent with our results on CelebA dataset in Section 4 of the main paper where we predict "Blonde Hair". More concretely, Undersampling performs comparably and sometimes better than nonsampling methods. This is reaffirming that predicting attributes on CelebA is relatively an easy task that dropping samples to balance subgroup distribution is sufficient to mitigate bias. However, as discussed in Section 4 of the main paper, vanilla sampling methods (Undersampling, Upweighting, Oversampling) perform poorly on some datasets. For example, as we note in Table 1 in the main paper, Undersampling performs considerably worse than nonsampling methods on the Utk-Face dataset as well the CIFAR-S dataset. Moreover, Upweighting performs substantially worse on CIFAR-S. Finally, Oversampling performs consistently worse on every benchmark. However, only our method, Bias Mimicking, manages to maintain competitive performance with respect to nonsampling methods on all datasets.

	US [13]	OS [35]	Other Methods
UTK-Face Age	400	7	20
UTK-Face Race	120	10	20
CelebA Blonde	170	4	10
CelebA Black Hair	40	5	10
CelebA Smiling	30	5	10
CIFAR-S	2000	100	200

Table 6. Number of Epochs used to train each method. Refer to Section D for further discussion.

D. Model and Hyper-parameters Details

We test bias Mimicking on six benchmarks. Three Binary Classification tasks on CelebA [22], namely, Blonde, Black Hair, and Smiling, Two Binary Classification tasks on UTK-Face [37], namely Race and Age and one multi-class task CIFAR-S. We provide further info below.

Optimization Following [12], we use ADAM [17] optimizer with learning rate 0.0001 on CelebA and UTK-Face. For CIFAR-S, following [35], we use SGD with learning rate 0.1. GroupDRO [27], however, has not been tuned before on the benchmarks in our study. Even for CelebA Blonde, the method was not tuned on the more challenging split in this study. Therefore, we grid search the learning rate/weight regularization/group adjustment and choose the best over the validation set. Refer to Table 5 for our final choices. With respect to BC+BB, the method was not benchmarked on CIFAR-S. Therefore, we run a grid search over the method's hyperparameters and choose $\alpha = 1.0$, $\gamma = 10$. Finally, as discussed in Section 4, UW struggles to optimize over CIFAR-S with learning rate 0.1. Therefore, we tune the learning rate and we find that 0.0001 to work the best over the validation set.

Total Number of Epochs As noted in Section 4.1 in the paper, a model trained with Undersampling sees fewer iterations than baselines per epoch and a model trained with Oversampling sees more iterations per epoch. Therefore, we adjust the number of epochs for both methods such that the total number of iterations seen by the model is the same across all methods tested in our experiments. Refer to Table 6 for a breakdown of the total number of epochs used to train each method.

Augmentations: For all benchmarks, we augment the input images with a horizental flip. BC+BB [12] uses extra augmentation functions. Refer to [12] for further details.

Splits: Note on CelebA, unlike [12], we use CelebA validation set for validation and test set for testing rather than using the validation set for testing and a split of the training set for validation.

	100%	75%	50%	25%
UTK-Face Age	**79.7**	78.9	78.0	76.7
UTK-Face Race	**90.8**	90.6	89.9	88.3
CelebA Blonde	**90.8**	90.3	90.1	89.9

Table 7. Comparing the performance of our method's Unbiased Accuracy (UA) where we use $x\%$ of the linear program solution. Refer to E for discussion.

E. Effect of the Linear Program Constraint

As discussed in Section 3 in the paper, We use a linear program to determine how the training distribution is subsampled to mimick the bias. The program is constrained such that the resulting distributions preserve the most number of samples because fewer retained samples may compromise the model performance. To verify this the importance of this step, we train our model using distributions that maintain x% of the Linear Program solution where $x < 100$. Note the results in Table 7. Note how the performance drops emphasizing the importance of this constraint.