BIAS MIMICKING: A SIMPLE SAMPLING APPROACH FOR BIAS MITIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

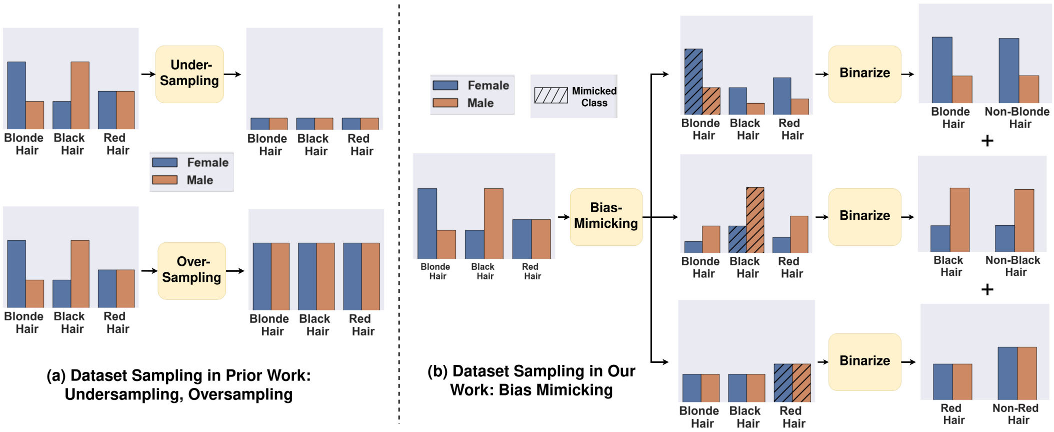
Prior work has shown that Visual Recognition datasets frequently under-represent sensitive groups (*e.g*. Female) within a category (*e.g*. Programmers). This dataset bias can lead to models that learn spurious correlations between class labels and sensitive attributes such as age, gender, or race. Most of the recent methods that address this problem require significant architectural changes or expensive hyper-parameter tuning. Alternatively, data re-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 hyperparameters, offer a cheaper and more efficient solution. However, we found that some of these baselines were missing from recent bias mitigation benchmarks. In this paper, we show that these simple methods are strikingly competitive with state-of-the-art bias mitigation methods on many datasets. Furthermore, we improve these methods by introducing a new class conditioned sampling method: Bias Mimicking. In cases where the baseline dataset re-sampling methods do not perform well, Bias Mimicking effectively bridges the performance gap and improves the total averaged accuracy of under-represented subgroups by over 3% compared to prior work.

1 INTRODUCTION

Spurious predictive correlations have been frequently documented within the Deep Learning literature (Wang et al., 2020a; Zhao et al., 2021). These correlations can arise when most samples in class y (*e.g*. blonde hair) belong to a sensitive group b (*e.g*. female). Thus, the model might learn to use the signal from b to predict y. Mitigating this spurious correlation (Bias) involves decorrelating the model's predictions of y from the dataset-sensitive group b. Previous research efforts have primarily focused on model-based solutions. These efforts can be mainly categorized into two directions 1) methods that require significantly more model parameters during inference (Wang et al., 2020b), which harms model scalability as we increase the number of sensitive groups/target-classes. 2) methods that introduce additional loss functions and require expensive hyper-parameter tuning (Hong & Yang, 2021; Kim et al., 2019a; Ryu et al., 2017; Tartaglione et al., 2021).

Dataset re-sampling methods, popular within the class imbalance literature (Japkowicz & Stephen, 2002; Buda et al., 2018; Sagawa et al., 2020; et al., 2018), present a simpler and cheaper alternative. Moreover, as illustrated in Figure 1(a), they can be easily extended to Bias Mitigation by considering the imbalance within the dataset subgroups rather than classes. Most common of these methods are Undersampling (Japkowicz & Stephen, 2002; Buda et al., 2018; Sagawa et al., 2020) and Oversampling (Wang et al., 2020b). They mitigate class imbalance by altering the dataset distribution through dropping/repeating samples, respectively. Another similar solution is Upweighting (Shimodaira, 2000; Byrd & Lipton, 2019) which levels each sample contribution to the loss function by appropriately weightings its loss value. However, these methods suffer from significant shortcomings. For example, Undersampling drops a significant portion of the dataset, which could harm models' predictive capacity. Moreover, Upweighting can be unstable when used with stochastic gradient descent (An et al., 2021). Finally, models trained with Oversampling, as shown by Wang et al. (2020b), 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), for each $y \in Y$, BM maintains a different version of the dataset target labels d_y. Each d_y preserves y samples while



(a) Dataset Sampling in Prior Work: Undersampling, Oversampling

(b) Dataset Sampling in Our Work: Bias Mimicking

Figure 1: Comparison of sampling-based approaches. **(a)** illustrates Undersampling/Oversampling methods that change the distribution of the dataset D such that $P_D(Y|B) = P_D(Y)$ by dropping samples/repeating samples, respectively, and thus mitigating the Bias of Y toward B. However, dropping samples hurt model predictive capacity, and repeating samples can cause overfitting with over-parameterized networks like neural nets Wang et al. (2020b). **(b)** shows our Bias Mimicking approach, where for each class y, our approach creates a different version of the class target labels d_y where the Bias of y is mimicked across the other classes. This process effectively decorrelates Y from B over dataset D (*i.e.* $P_D(Y|B) = P_D(Y)$). We then binarize each d_y and train a binary classifier to predict y. By using all the d_y(s) collectively, we expose the model to all the samples in D. Moreover, since we do not repeat samples in each d_y, our method is less prone to overfitting

Undersampling $y' \neq y$ such that the bias within class y' *mimics* that of y. Consequently, for each d_y, Y is statistically independent from B, *i.e.* $P_{d_y}(Y = y|B = b) = P_{d_y}(Y = y)$. A naive way of using each d_y is to dedicate a different multi-class prediction head for each. However, this could present scalability issues as the number of classes/sensitive-groups increases. Alternatively, we binarize each d_y. Then, we train a different one-vs-all binary classifier on each d_y. BM addresses the shortcomings of prior work's sampling methods. For example, using every d_y to train the model exposes it to the entire dataset, whereas Undersampling can discard significant numbers of samples. Our approach also does not use weights to scale the loss function. Thus our method should avoid stability issues using stochastic gradient descent suffered by Upweighting. Finally, since each binary predictor is not exposed to repetitive samples, our model is less prone to overfitting.

In addition to proposing Bias Mimicking, another contribution of our work is providing an extensive analysis on sampling methods for bias mitigation tasks. We found many sampling-based methods were notably missing in the comparisons used in prior work (Tartaglione et al., 2021; Wang et al., 2020b; Hong & Yang, 2021). 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, for cases where these methods are not as effective, Bias Mimicking effectively bridges the performance gap and improves over prior work model-based methods. Finally, we thoroughly analyze our approach's behavior through two experiments. First, it is unclear how sensitive our method is to the mimicking condition. Therefore, in Section 4.3, we simulate various scenarios where the Bias is not perfectly mimicked and note model performance. Second, we verify the importance of each d_y to the method predictive performance in Section 4.4. Both experiments showcase the importance of our design in mitigating Bias.

Our contributions can be summarized as:

- We show that simple resampling methods are be competitive on many benchmarks when compared to expensive model-based state-of-the-art approaches.
- We introduce a novel resampling method: Bias Mimicking that improves the average accuracy over under-represented subgroups by 3% over multiple datasets.
- We conduct an extensive empirical analysis of Bias Mimicking that details the method's sensitivity to the Mimicking condition and uncovers various insights about its behavior.

2 Related Work

Documenting Spurious Correlations: Bias in Machine Learning can manifest in many ways. Examples include class imbalance Japkowicz & Stephen (2002), historical human biases Suresh & Guttag (2019), evaluation bias et al. (2018), and more. For a full review, we refer the reader to Mehrabi et al. (2021). In our work, we are interested in model bias that arises from spurious correlations. A spurious correlation results from under-representing a certain group of samples (*e.g.* samples with 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, Singh et al. (2020); Hendrycks et al. (2021); Xiao et al. (2020); Li et al. (2020) showed that state-of-the-art object recognition models are biased toward backgrounds or textures associated with the object class. Agrawal et al. (2018); Clark et al. (2019) showed similar spurious correlations in VQA. Zhao et al. (2021) noted concerning correlations between captions and attributes like skin color in Image-Captioning datasets. Beyond uncovering these correlations within datasets, prior work like Wang et al. (2020b); Hong & Yang (2021) introduced synthetic bias datasets where they systematically assessed bias effect on model performance.

Model based solutions: In response to documentation of dataset spurious correlations, several model focused methods have been proposed (Ryu et al., 2017; Kim et al., 2019a; Wang et al., 2020b; Hong & Yang, 2021; Tartaglione et al., 2021). For example, Tartaglione et al. (2021) presents a metric learning-based method where model feature space is regularized against learning harmful correlations. Wang et al. (2020b) surveys several existing methods such as adversarial training, that randomizes the relationship between target classes and sensitive groups in the feature space. They also present a new method, domain-independence, where different prediction heads are allocated for each sensitive group. Most recently, Hong & Yang (2021) extended contrastive learning frameworks on self-supervised learning (Chen et al., 2020; He et al., 2020; Khosla et al., 2020) to mitigate bias. Our work complements these efforts by introducing a hyper-parameter-free re-sampling algorithm that improves over state-of-the-art performers.

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 and Undersampling majority ones, which are popular within the class imbalance literature (Japkowicz & Stephen, 2002; Buda et al., 2018; Sagawa et al., 2020; et al., 2018). However, as we note in our introduction, some of these methods have been missing in recent visual Bias Mitigation Benchmarks (Wang et al., 2020b; Tartaglione et al., 2021; Hong & Yang, 2021). Thus, we review these methods and describe their shortcomings in Section 3.1. Alternatively, other efforts attempt to fix the dataset distribution by introducing completely new samples. Examples include (Kärkkäinen & Joo, 2021) where they introduce a new dataset for face recognition that is balanced among several race groups, and Ramaswamy et al. (2020) where they used GANs to generate training data that balance the sizes of dataset subgroups. While a dataset-based approach, our work differs from these efforts as it does not generate or introduce new samples. Finally, also related to our work are sampling methods like REPAIR (Li & Vasconcelos, 2019) where a function is learned to prioritize specific samples and, thus, learn more robust representations. However, unlike our method, REPAIR does not make use of sensitive group labels and thus is not able to target specific spurious correlations.

3 Sampling For Bias Mitigation

In bias mitigation the goal is to remove the effect of spurious correlations to prevent a model from correlating its predictions with sensitive groups of samples. More formally, assume we have a dataset of image/target/sensitive-group triplets (X, Y, B). Define $g_{y,b} = \{(x_i, y_i, b_i) \text{ s.t } y_i = y, b_i = b\}$, *i.e.* the *subgroup* of samples that share the target class y and sensitive group b. Spurious correlations occur when the samples of a target class y (*e.g.* blonde hair) are over-represented by one sensitive group b (*e.g.* female) rather than distributed equally among the dataset sensitive groups. In other words, assuming $P_D(X, Y, B)$ is the distribution of the training data, then $P_D(B = b, Y = y) >> \frac{1}{|B|}$ where $|B|$ denotes the number of sensitive groups. Consequently, the model may use the signal from B to predict Y. For example, if most blonde hair samples were female, the model might learn to predict blonde hair every time it sees a female sample. Thus, the goal of this task is to train the model such that $P(\hat{Y}|X, B) = P(\hat{Y}|X)$ where $\hat{Y}$ denotes model predictions.

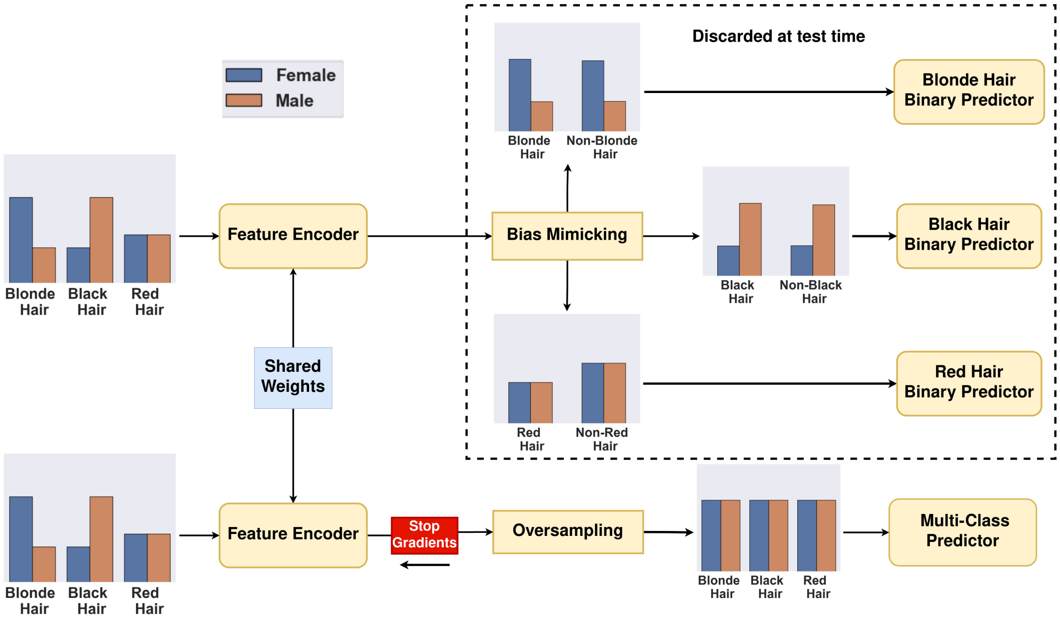


Figure 2: **Bias Mimicking** takes in the target labels distribution and produces a version for each class y (*i.e.* Black, Blonde and Red Hair). Each version, denoted as d_y, is a binary distribution where the positive class y is preserved and the negative class combines every $y' \neq y$ where the bias of y is mimicked across every y', thus $P_D(Y|B) = P_D(Y)$ following Proposition 1. Using every d_y mitigates the bias within the feature space while exposing the model to the entire distribution. Finally, during inference, a multi-class prediction head is trained on top of learned featured space. As we note in Section 3.2 Oversampling the input features when training the prediction head results in a slight but significant boost on some benchmarks. Furthermore, we ensure that the gradients do not propagate to the feature space to prevent the neural net from overfitting over repetitive samples.

Our work addresses methods that fix spurious correlations by ensuring statistical Independence between Y and B on the dataset level, *i.e.* $P_D(Y|B) = P(Y)$. As we note in our introduction, some of the solutions that fall under this approach are missing from prior work benchmarks. Therefore, We briefly review the missing methods in Section 3.1. Then, we introduce a new sampling method: Bias Mimicking which addresses prior work sampling methods shortcomings in Section 3.2.

3.1 Simple Sampling Methods

As discussed in the introduction, the class imbalance literature is rich with dataset re-sampling solutions (Japkowicz & Stephen, 2002; Buda et al., 2018; Sagawa et al., 2020; et al., 2018; Chawla et al., 2002). These solutions address class imbalance by balancing the distribution of classes. Thus, they can be extended to Bias Mitigation by balancing subgroups instead of classes. Prior work in Visual Bias Mitigation has explored one of these solutions: Oversampling (Wang et al., 2020b). This approach mitigates dataset bias by replicating the samples of the underrepresented subgroups until all subgroups are balanced. However, Wang et al. (2020b) has demonstrated that Oversampling is not effective at mitigating bias, likely because the model sees repetitive sample copies, thus resulting in overfitting. In addition to Oversampling, Undersampling (Japkowicz & Stephen, 2002; Buda et al., 2018; Sagawa et al., 2020) and Upweighting (Shimodaira, 2000; Byrd & Lipton, 2019) are other popular methods. Both methods, however, have not been benchmarked in recent Visual Bias Mitigation work (Tartaglione et al., 2021; Hong & Yang, 2021). We review these solutions below.

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. More concretely, we subsample every subgroup in the dataset to match the size of the smallest subgroup, *i.e.* $\min_{y,b} |g_{y,b}|$. However, a critical shortcoming of Undersampling is that it drops a significant portion of the dataset and thus could harm the model's predictive capacity. This might explain its absence from recent bias mitigation benchmarks. We address this shortcoming in our proposed method, Bias Mimicking, in Section 3.2.

Upweighting Upweighting levels the contribution 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 y, and subgroup $g_{y,b}$ where $x \in g_{y,b}$, then the model optimizes:

$$L = E_{x,y,g}\Big[\frac{1}{p_{g_{y,b}}} l(x, y; w)\Big]$$

where $p_{g_{y,b}} = \frac{|g_{y,b}|}{\sum_{y,b} |g_{y,b}|}$ computed over the training dataset. A key shortcoming of Upweighting is its instability when used with stochastic gradient descent (An et al., 2021). Indeed, we demonstrate this problem in our experiments where Upweighting does not work well on some datasets.

3.2 Bias Mimicking

The goal in our method is to decorrelate a model's predictions $\hat{Y}$ from sensitive attributes B. Our approach is inspired by Sampling methods, which enforce this independence on the dataset level (*i.e.*$P_D(Y|B) = P_D(Y)$). However, simple sampling methods like Oversampling, Undersampling, and Upweighting suffer from critical shortcomings as outlined in Section 3.1. We address these shortcomings in our proposed sampling algorithm: Bias Mimicking below.

Algorithm The key principle of our algorithm is the observation that if bias toward sensitive group b was proportionally equal among all the target classes, then for every $y \in Y$, $P(Y = y|B = b) = P(Y = y)$, *i.e.* y is statically independent from b. More concretely:

Proposition 1. *Assume target labels set* $Y = \{0, 1, 2, ..., C\}$ *and sensitive group* $b \in B$*, if* $P(B = b|Y = i) = P(B = b|Y = j) \quad \forall i, j \in \{0, ..., C\}$*, then* $P(Y = y|B = b) = P(Y = y) \quad \forall y \in Y$*.*

Refer to Appendix A for proof. Note that ensuring this proposition holds for every $b \in B$ implies that $P(Y|B) = P(Y)$, in other words Y is independant from B which in turns means that the resulting model's predictions will not be strongly correlated with B. Refer to Section 4.3 for an empirical sensitivity analysis of this result. Note how under-sampling is a special case of this result where indeed Y is independent of B.

We use this result to motivate a novel way of Undersampling the input distribution that ensures the model is exposed to every sample in the dataset while preventing a spurious correlation. To that end, for every class y, we create a different version of the dataset target labels. Denote each version as d_y. Each d_y preserves its respective class y samples while Undersampling every $y' \neq y$ such that

$$P_{d_y}(B = b|Y = y) = P_{d_y}(B = b|Y = y') \quad \forall b \in B \tag{1}$$

As a result, each d_y ensures the independence of Y from B. A naive way of using each d_y would be to dedicate a multi-class prediction head for each. However, this could present scalability issues as the number of classes/sensitive-groups increases. Alternatively, We binarize each d_y and then use it train a one-vs-all binary classifier BP$_y$ for each y. Each head is trained on image-target pairs from its corresponding distribution d_y as Figure 2 demonstrates. Consequently, since each d_y preserves y samples, the model backbone sees the entire dataset through the different signals backpropagating from each BP$_y$. Moreover, since the same bias is mimicked for every class in d_y, each incoming signal from BP$_y$ does not backpropogate spurious correlations, thus, Y correlation with B should be minimized within the model learned feature representation.

Using the binary classifiers during inference is challenging since each was trained on different distributions and are, therefore, uncalibrated compared to each other. However, we know that 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 while preserving the original dataset distribution. Note that we stop the gradient from flowing into the model backbone to ensure that the bias is not learned again. In our experiments, we found that this achieves state-of-the-art performance on many benchmarks. However, on some, we found that we could obtain a slight improvement in performance by Oversampling the distribution of the features used to train the multi-class prediction head. Oversampling, unlike when used to train the entire model, did not overfit in this setting because 1) the multi-class prediction head is a simple linear layer which is less likely to overfit than an overparameterized model like a neural net 2) the features learned by our

Table 1: **Binary classification benchmark** compare methods LNL (Kim et al., 2019b), EnD (Tartaglione et al., 2021), DI (Wang et al., 2020b), BC+BB (Hong & Yang, 2021), Undersampling (US) (Japkowicz & Stephen, 2002), Upweighting (UW) (Byrd & Lipton, 2019), and Bias Mimicking (BM, ours), on the CelebA and UTK-face dataset. Methods are evaluated using Unbiased Accuracy (Hong & Yang, 2021) (UA), Bias-conflict (Hong & Yang, 2021) (BC), as well as the average of each metrics over both datasets. *Methods in italic* either introduces additional hyper-parameters or model parameters. See Section 4.1 for discussion.

	UTK-Face				CelebA		All Datasets		Additional	
Method	Age		Race		Blonde		Average		Hyper Param	Model Param
	UA	BC	UA	BC	UA	BC	UA	BC		
Vanilla	72.0±0.3	45.3±0.5	88.1±0.1	80.8±0.5	78.7±0.6	58.4±1.2	79.6±0.3	61.5±0.7		
LNL	72.9±0.3	47.0±0.2	89.1±0.2	81.2±0.1	80.3±0.3	61.8±0.8	80.7±0.2	63.3±0.3	True	
DI	75.5±1.1	58.8±3.6	90.7±0.1	90.9±0.4	90.9±0.5	86.1±0.8	85.7±0.5	77.8±1.6		True
EnD	73.1±0.6	46.9±0.7	88.5±0.4	81.3±0.3	86.5±1.4	75.8±1.8	82.7±0.8	68.0±0.9	True	
BC+BB	79.4±0.7	72.2±4.0	91.4±0.2	90.6±0.5	91.2±0.2	87.2±0.5	**87.3±0.3**	83.3±1.6	True	
OS	77.0±1.3	63.2±5.4	90.8±0.6	87.9±0.9	82.4±2.3	66.8±5.2	83.4±1.4	72.6±3.8		
UW	78.6±1.1	76.8±3.9	90.3±0.2	90.1±0.5	91.6±0.3	88.3±0.5	86.8±0.5	85.0±1.6		
US	68.9±2.1	70.4±3.5	88.1±0.6	89.3±0.5	91.1±0.2	89.1±0.4	82.7±0.9	82.9±1.1		
BM (ours)	79.8±1.2	80.5±2.6	90.9±0.4	91.1±0.6	91.0±0.2	87.3±0.3	**87.2±0.5**	**86.3±1.1**		

method do not take gradients from the oversampled approach, helping to ensure that the neural net can not overfit over the reptitive samples. Refer to Appendix B for further analysis. Given these results, we use Oversampling to train the multi-class prediction head throughout our experiments.

Cost Analysis Unlike prior work (Hong & Yang, 2021; Tartaglione et al., 2021), bias mimicking involves no extra hyper-parameters. The debiasing is automatic by definition. Moreover, unlike prior work (Wang et al., 2020b) where an that used each prediction head trained on every protected attribute (*i.e.* $O(B)$), bias mimicking uses only a single prediction head during inference (*i.e.* $O(1)$), discarding the attribute specific prediction heads at test time. Therefore, our method is simpler and more efficient during inference. Finally, note that the additional target labels d_y 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) accordingly.

4 Experiments

We report our method performance on two benchmarks: a binary classification benchmark in Section 4.1 and a multi-class classification benchmark in Section 4.2. In addition to reporting our method results, we expand both benchmarks, which mainly focus on model-based methods, by including the basic sampling methods outlined in Section 3, namely Undersampling, Upweighting, and Oversampling. Then, we follow up our results with two main experiments that analyze our method's behavior. The first experiment in Section 4.3 is a sensitivity analysis of our method to the mimicking condition. The second experiment in Section 4.4 analyzes the contribution of each version of the target labels d_y to model performance.

4.1 Binary Classification Benchmark

Datasets and Metrics We compare methods using CelebA dataset (Liu et al., 2015) and UTKFace dataset (Zhang et al., 2017). Following prior work (Hong & Yang, 2021; Tartaglione et al., 2021), we train a binary classification model using CelebA where BlondHair is a target attribute and Gender is a bias attribute. Note that prior work (Hong & Yang, 2021; Tartaglione et al., 2021) used the HeavyMakeUp attribute in their CelebA benchmark. However, during our experiments, we found serious problems with the benchmark. Refer to Appendix C for model details. Therefore, we skip

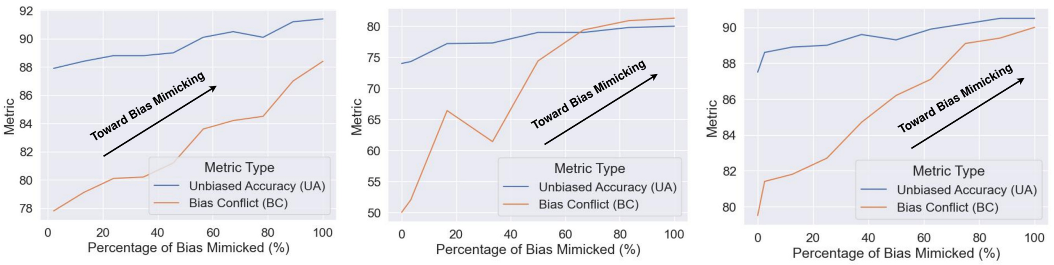
Table 2: **Multi-Class classification benchmark** Compare methods: Adv w/ uniform confusion (Hoffman et al., 2015) and reversal projection (Zhang et al., 2018), DS (Wang et al., 2020b) w/ RBA (Zhao et al., 2017), DI (Wang et al., 2020b), Undersampling (US) (Japkowicz & Stephen, 2002), Upweighting (UW) (Byrd & Lipton, 2019), Oversampling (OS) (Wang et al., 2020b) and Bias Mimicking (BM, ours), on the Cifar-s dataset (Wang et al., 2020b). Methods in *italic* either introduces hyper-parameters or model-parameters. Note that N below denotes the number of classes while D denotes the number of sensitive groups. Refer to (Wang et al., 2020b) for further details on the variations of model based methods below. See Section 4.2 for results discussion.

						Additional	
Model Name	Model	Bias Amp ↓	Color Acc ↑	Gray Acc ↑	Mean Acc ↑	Hyper Params	Model Params
Vanilla	N-way softmax	0.074	89.0	88.8	88.5±0.3		
Adv	w/ uniform confusion	0.101	83.8	83.9	83.8±1.1	True	
	w/ ∇ reversal, proj	0.09	84.6	83.5	84.1±1.0	True	
DS	Joint ND-way softmax	0.040	91.2	89.4	90.3±0.5		True
	RBA	0.054	89.2	88.0	88.6±0.4		True
DI	Separate ND-way softmax	0.004	**92.4**	**91.7**	**92.0±0.1**		True
OS	N-way softmax	0.066	89.2	89.1	89.1±0.4		
UW	N-way softmax	0.004	84.7	85.0	84.8±0.2		
US	N-way softmax	**0.003**	70.8	68.9	68.8±1.1		
BM (ours)	N-way softmax	0.004	92.0	**91.6**	**91.8±0.2**		

this benchmark in our work. For UTKFace, we follow Hong & Yang (2021) and do the binary classification with Race/Age as the sensitive attribute and Gender as the target attribute. Methods are evaluated in terms of Unbiased Accuracy (Hong & Yang, 2021) which measures the accuracy on a test set balanced among each subgroup, and Bias-Conflict (Hong & Yang, 2021) which measures the accuracy on the minority subgroups. Refer to Appendix D for more details.

Baselines We use the baselines reported in prior work (Hong & Yang, 2021) benchmark. These methods either require additional model parameters or hyper-parameters. "Bias-Contrastive and Bias-Balanced Learning" (BC + BB) (Hong & Yang, 2021) uses a contrastive learning framework to mitigate bias. "Learning Not to Learn" (LNL) uses an additional prediction branch to minimize the mutual information between the model feature and the bias label. Domain-independent (DI) (Wang et al., 2020b) uses an additional prediction head for each sensitive subgroup. EnD (Tartaglione et al., 2021) uses a regularizer that disentangles the features of the same bias class samples. Furthermore, we expand the benchmark by reporting the performance of sampling methods outlined in Section 3. All reported methods are compared to a "vanilla" model trained on the original dataset distribution with a Cross-Entropy loss.

Results in the binary classification benchmark in Table 1 indicates that our method (BM) maintains strong averaged predictive accuracy (UA) while significantly improving performance over the under-represented subgroups (BC) by $> 3\%$ when compared to prior work model-based methods. Moreover, our method's strong performance does not require additional expensive hyper-parameter tuning or model parameters, unlike prior work methods. Therefore, our method is simpler and more efficient. Furthermore, note that our method performs the best among the sampling methods. More concretely, BM performs significantly better than Oversampling. This aligns with (Wang et al., 2020b) observation that overparameterized models like neural nets tend to overfit when trained on Oversampled distributions. Note that even though our method uses Oversampling to train part of the network, we do not see the same decline in performance. This is because, as discussed in Section 3.2, we only use the oversampled version of the dataset to train a simple linear layer over the learned debiased features. Undersampling demonstrates poor predictive performance on UTK-Face due to Undersampling a significant portion of the dataset, which is then reflected in the average performance. Surprisingly, however, the method performs quite well on CelebA. This is likely because the task (predicting hair color) is easy and does not require many samples. Thus, this emphasizes the method's importance as a reasonable first step in case sufficient data is available. Overall, however, the average performance of our method is significantly better. Finally, note that while Upweight-



a) CelebA/Blonde

b) Utk-Face/Age

a) Utk-Face/Race

Figure 3: **Model Sensitivity to Bias Mimicking** We test our method's sensitivity to the Bias Mimicking condition in Equation 1. 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 tasks introduced by the Binary Classification Benchmark in Section 4.1. Refer to Section 4.3 for discussion.

ing demonstrates competitive performance on CelebA, it lags behind significantly over Utk-Face. This is likely due to the method's instability with respect to stochastic gradient descent (An et al., 2021). Thus, when comparing the method's averaged performance to ours, we note that our method improves performance by over 1%.

4.2 Multi-Class Classification Benchamrk

Datasets and Metrics We use the bias controlled CIFAR-S benchmark (Wang et al., 2020b). The benchmark synthetically introduces bias into the CIFAR-10 dataset (Krizhevsky, 2012) by converting a sub-sample of images from each target class to gray-scale. We use (Wang et al., 2020b) 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 represent 95% of the samples. Methods are evaluated using the accuracy on two versions of the test sets: color and gray as well as the mean of both sets. In addition to accuracy, methods are evaluated with the bias amplification metric (Wang et al., 2020b).

Baselines We use the baselines reported in prior work (Wang et al., 2020b) benchmark. These methods either require additional model parameters or hyper-parameters. Adversarial Learning (Adv) w/ uniform confusion (Hoffman et al., 2015) and reversal projection (Zhang et al., 2018), introduces an adversarial loss that seeks to randomize model's feature representation of Y and B, Domain Discriminative training (Wang et al., 2020b) and Domain Independence (DI) (Wang et al., 2020b) both introduce a prediction head for each dataset subgroup. However, they use different methods to train and perform inference. Refer to (Wang et al., 2020b) for more information. Furthermore, we expand the benchmark by including the sampling methods outlined in Section 3. All reported methods are compared to a ”vanilla” model trained on the original dataset distribution with a Cross-Entropy loss.

Results Observe the results in Table 2. Note that our method (BM) is able to maintain competitive performance over prior work state of the art (DI) while requiring no additional hyperparameters or model parameters at inference time. Moreover, when considering DI poor performance on the Binary Classification benchmark in Table 1, our method averaged performance with the results from Table 2 is significantly better. More notably, it is the only sampling method that maintains both strong predictive performance and significantly reduces the bias. Upweighting weaker predictive performance here can be attributed to its instability with respect to stochastic gradient descent (An et al., 2021). While Undersampling reduces bias significantly compared to prior work, it lags significantly with its predictive performance.

4.3 How sensitive is the model performance to the Mimicking condition?

Bias Mimicking is effective at mitigating Bias. However, it is not clear how sensitive the model's learned spurious predictive behavior is to the Bias mimicking condition (Equation 1). Therefore, in this section, we seek to answer this question. To that end, we test multiple scenarios where the Bias is mimicked by a percentage value $x \in \{0, 100\}$ such that 0% corresponds to *no* modification to the distribution and 100% corresponds to complete Bias mimicking. Following this definition,

Table 3: We investigate the effect of each re-sampled version of the dataset d_y on model performance using the binary classification tasks outlined in Section 4.1. 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 each class's relevant subgroups only. Refer to section 4.4 for discussion

(a) Utk-Face/Age

	UA$_1$	UA$_2$	UA
(d_1)	82.5	74.2	78.4±0.5
(d_2)	73.1	67.9	70.5±2.3
(d_1, d_2)	84.4	75.3	**79.8±0.9**

(b) Utk-Face/Race

	UA$_1$	UA$_2$	UA
(d_1)	90.7	85.1	87.9±0.5
(d_2)	84.8	91.5	88.1±0.6
(d_1, d_2)	90.5	91.1	**90.9±0.4**

(c) CelebaA/Blonde

	UA$_1$	UA$_2$	UA
(d_1)	91.5	90.3	90.9±0.1
(d_2)	82.2	96.5	89.3±0.1
(d_1, d_2)	88.1	94.2	**91.0±0.2**

we run the experiment on the datasets and attributes introduced by the Binary Classification Benchmark in Section 4.1 and evaluate performance using the metrics introduced in Section 4.1, namely Bias Conflict (BC) and Unbiased Accuracy (UA). 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 $P(Y|B) \neq P(Y)$ following proposition 1. The best performance is achieved when $x\%$ is indeed 100% and thus $P(Y|B) = P(Y)$. From this analysis, we conclude that the Bias Mimicking condition is critical for good performance.

4.4 How does each d_y affect model performance?

Bias Mimicking takes in the original dataset distribution of target labels and produces a different version for each class y, denoted as d_y, where class y samples are preserved, and the bias of y is mimcked in the other classes. By using every d_y, we expose the model to the entire dataset. In this section, we investigate the effect of each d_y on performance. To that end, we perform an experiment using the binary classification tasks outlined in Section 4.1. For each task, thus, we have two versions of the dataset target labels d_1, d_2. We compare three different models performances: model (1) trained on only (d_1), model (2) trained on only (d_2) and finally model (3) trained on both (d_1, d_2). The last version being 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. Note that, overall, the model trained on (d_1) performs better at predicting y_1 but worse at predicting y_2 as outlined by results in Table 3. We note the same trend with the model trained on (d_2) but in reverse. This disparity in performance harms the Unbiased Accuracy. However, a model trained on (d_1, d_2) balances both accuracies and achieves the best total Unbiased Accuracy (UA). These results emphasize the importance of each d_y for good performance.

5 Conclusion

We first recognized that simple and hyper-parameter-free methods for bias mitigation like Undersampling and Upweighting were missing from recent benchmarks. This observation was especially relevant considering that state-of-the-art solutions require either expensive hyper-parameter tuning or architectural change. Therefore, we benchmarked these methods and concluded that some are surprisingly effective on many datasets. However, on some others, their performance significantly lagged behind model-based methods. Motivated by this observation, we introduced a novel sampling method: Bias Mimicking. The method retained the simplicity of sampling methods while improving over state-of-the-art model-based methods. Furthermore, we extensively analyzed the behavior of Bias Mimicking, which emphasized the importance of our design.

Future Work We demonstrate 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. These are all questions that we hope future work could address.

Code of Ethics Statement Our work addresses a critical problem within the fairness literature: spurious predictive behavior by Deep Learning models. We measure this behavior by calculating the dataset's subgroups' predictive performance (accuracy). While this metric aligns with our goal of measuring and preventing spurious behavior, we emphasize that the metric is not exhaustive of other fairness concerns. We refer the reader to (Kleinberg et al., 2017) for a broader discussion of fairness metrics. Furthermore, while our proposed method aims to learn robust representations for underrepresented subgroups within a given dataset, we acknowledge that such representations could be misused in downstream applications that could raise ethical concerns (*e.g.* surveillance). Furthermore, two of the datasets used in our experiments involve facial attribute recognition tasks (*e.g.* hair color). Models trained on these datasets could be misused, yet they remain standard benchmarks for evaluating spurious correlations. We encourage future researchers and practitioners to use this technology with care and caution.

Reproducibility Statement We provide the source code for this work in the supplementary. The training and hyper-parameters to reproduce the results are outlined in Appendix D. Moreover, The results we report in this paper were averaged over 3 runs with different seeds, and standard deviations were reported to ensure statistical significance. Furthermore, all experiments were done on datasets that are publicly available online.

REFERENCES

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*, pp. 4971–4980, 2018.

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. URL https://openreview.net/forum?id=iQQK02mxVIT.

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. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2018.07.011. URL https://www.sciencedirect.com/science/article/pii/S0893608018302107.

Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 872–881. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/byrd19a.html.

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. doi: 10.1613/jair.953.

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.

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)*, pp. 4069–4082, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1418. URL https://aclanthology.org/D19-1418.

Joy Buolamwini et al. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *ACM FAccT*, 2018.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

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)*, pp. 9726–9735, 2020. doi: 10.1109/CVPR42600.2020.00975.

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

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

Youngkyu Hong and Eunho Yang. Unbiased classification through bias-contrastive and bias-balanced learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=2OqZZAqxnn.

Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, pp. 429–449, 2002.

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 (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

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)*, pp. 9004–9012, 2019a. doi: 10.1109/CVPR.2019.00922.

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 2019b.

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

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Innovations in Theoretical Computer Science*, 2017.

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

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)*, pp. 1547–1557, 2021. doi: 10.1109/WACV48630.2021.00159.

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

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.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

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. ISSN 0360-0300. doi: 10.1145/3457607. URL https://doi.org/10.1145/3457607.

Vikram V. Ramaswamy, Sunnie S. Y. Kim, and Olga Russakovsky. Fair attribute classification through latent space de-biasing. *CoRR*, abs/2012.01469, 2020. URL https://arxiv.org/abs/2012.01469.

Hee Jung Ryu, Margaret Mitchell, and Hartwig Adam. Improving smiling detection with race and gender diversity. *CoRR*, abs/1712.00193, 2017. URL http://arxiv.org/abs/1712.00193.

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 (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8346–8356. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/sagawa20a.html.

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. doi: 10.1016/S0378-3758(00)00115-4.

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.

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

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)*, pp. 13508–13517, June 2021.

Angelina Wang, Arvind Narayanan, and Olga Russakovsky. Vibe: A tool for measuring and mitigating bias in image datasets. *CoRR*, abs/2004.07999, 2020a. URL https://arxiv.org/abs/2004.07999.

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)*, 2020b.

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.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, pp. 335–340, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360128. doi: 10.1145/3278721.3278779. URL https://doi.org/10.1145/3278721.3278779.

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

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

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2979–2989, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1323. URL https://aclanthology.org/D17-1323.

A Propositions and proofs

The central component of our proposed method is the insight that if the the bias toward sensitive group b was proportionally equal among every $y \in Y$, then $P(Y = y|B = b) = P(Y = y)$. We provide a proof for this proposition below:

Proposition 1. *Assume target labels set* $Y = \{0, 1, 2, ..., C\}$ *and sensitive group* $b \in B$*, if* $P(B = b|Y = i) = P(B = b|Y = j) \quad \forall i, j \in \{0, ..., C\}$*, then* $P(Y = y|B = b) = P(Y = y) \quad \forall y \in Y$*.*

Table 4: **Sampling for multi-class prediction head-Binary Benchmark** compare the effects of using different sampling methods to train the multi-class prediction head to perform inference in our proposed method: Bias Mimicking. Refer to Section B for discussion.

	UTK-Face				CelebA		All Datasets	
Method	Age		Race		Blonde		Average	
	UA	BC	UA	BC	UA	BC	UA	BC
Vanilla	72.0±0.3	45.3±0.5	88.1±0.1	80.8±0.5	78.7±0.6	58.4±1.2	79.6±0.3	61.5±0.7
BM + Vanilla	80.0±1.0	80.3±1.4	90.7±0.3	90.8±0.5	90.5±0.6	85.9±1.3	87.0±0.6	85.6±1.0
BM + UW	79.8±0.8	79.3±3.4	90.8±0.4	91.3±0.5	91.3±0.2	87.5±0.5	**87.2±0.4**	**86.0±1.4**
BM + US	79.8±0.8	79.7±3.4	90.5±0.3	90.7±0.6	90.9±0.3	87.1±0.9	**87.1±0.4**	**85.8±1.6**
BM + OS	79.8±1.2	80.5±2.6	90.9±0.4	91.1±0.6	91.0±0.2	87.3±0.3	**87.2±0.5**	**86.3±1.1**

Table 5: **Sampling for multi-class prediction head-Multi class Benchmark** compare the effects of using different sampling methods to train the multi-class prediction head to perform inference in our proposed method: Bias Mimicking.

Model	Bias Amp ↓	Color Acc ↑	Gray Acc ↑	Mean Acc ↑
Vanilla	0.074	89.0	88.8	88.5±0.3
BM + Vanilla	0.004	91.8	91.3	91.5±0.1
BM + US	**0.001**	19.8	19.7	19.7±0.7
BM + UW	0.005	91.7	91.2	91.4±0.1
BM + OS	0.004	**92.0**	**91.6**	**91.8±0.1**

Proof. First, given the law of total probability:

$$P(B=b) = \sum_{y' \in Y} P(B=b|Y=y')P(Y=y')$$

Given our assumption of bias mimicking, we can write $\forall y \in Y$:

$$P(B=b) = P(B=b|Y=y) \sum_{y' \in Y} P(Y=y') = P(B=b|Y=y) \tag{2}$$

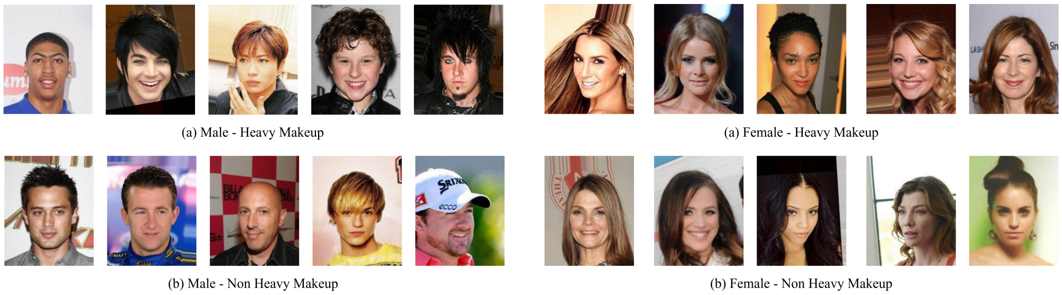
From here, using Bayesian probability and the result from (2), we can write, $\forall y \in Y$:

$$P(Y=y|B=b) = \frac{P(B=b|Y=y)P(Y=y)}{P(B=b)} = \frac{P(B=b)P(Y=y)}{P(B=b)} = P(Y=y)$$

□

B SAMPLING METHODS IMPACT ON MULTI-CLASS CLASSIFICATION HEAD

Bias Mimicking produces a binary version d_y of the labels for each class y. Each d_y preserves class y samples while undersampling each y' such that the bias within y' mimics y. A debiased feature representation is then learned by training a binary classifier for each d_y. When the training is done, it is challenging to use the scores from each binary predictor for inference. 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 on top of the learned feature representations using the original dataset distribution, where we prevent the gradients from flowing into the feature space. Observe that this approach denoted by BM + Vanilla in Table 4 and Table 5 is effective at mitigating the bias when compared to vanilla model results. However, we note that we obtained a slight performance improvement when we used Oversampling mostly on the CelebA dataset in Table 4 and on the Cifar-s dataset in Table 5. Therefore, we use Oversampling to train the multi-class prediction head in our implementation.


(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. Note the that there is not a clearly differentiating signal for the attribute Heavy Makeup.

Table 6: Label Noise in Heavy Makeup attribute

	Non-Heavy Makeup	Heavy Makeup
Female	34%	25%
Male	9%	20%

Aside from Oversampling, note that other sampling methods, namely Undersampling and Upweighting, could not demonstrate the same improvement in performance. Both were comparable to Oversampling on the Binary Benchmark in Table 4. However, both lagged significantly behind the multi-class benchmark in Table 5.

C Heavy Makeup Benchmark

Prior work Hong & Yang (2021) uses the Heavy Makeup binary attribute prediction task from CelebA [20] 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. In our experiments, we found that this benchmark contains significant noise. We believe this noise stems from the fact that Heavy Makeup is subjective metric. Moreover, It is influenced by cultural elements, lighting conditions, and camera pose. Thus, we expect a fair amount of label noise, which we verify via a qualitative and quantitative analysis below.

Qualtiative 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.

Quantitative Analysis: We sample 100 random images from the following subgroups: Female-Heavy Makeup, Female-Non-Heavy Makeup, Male-Heavy Makeup, and Male-Non-Heavy Makeup. We calculated the percentage of noisy images for each subgroup, *i.e.* images that are not clear whether they correctly belong to their subgroup. Observe the results in Table 6. Note that there is a significant amount of noise in each subgroup. Furthermore, the noise is amplified for the subgroups that involve the sensitive group: Female. The noise is further amplified when the test set used in Hong & Yang (2021) is examined. The test set for Male-Heavy Make up (an under-represented subgroup) only contains 9 samples. we could not visually determine whether 4 out of these 9 images fall under Heavy MakeUp. Out of the 5 left images, 3 are images of the same person from different angles. Therefore, given the noise in the training set, the small size of the under-represented group in the test set, and its noise, we conclude that results from this benchmark will be pretty noisy. Therefore, we choose to skip it in our experiments.

D Model and Hyper-parameters Details

We test our method (Bias Mimicking) on two benchmarks. The first benchmark is Binary Classification Benchmark that includes two datasets: CelebA (Liu et al., 2015), UTK-Face (Zhang et al., 2017) as outlined in Section 4.1. We follow the same training procedure as prior work (Hong & Yang, 2021). For both datasets, we train a Resnet-18 (He et al., 2016) model. We train the model for a total of 10 epochs on CelebA and 20 epochs on UTK-Face. For both datasets, we use a learning rate of $1e - 3$ with ADAM (Kingma & Ba, 2014) optimizer. We decay the learning rate following an exponential schedule at epochs 3 and 6 for CelebA and 7 and 14 for UTK-Face with $\gamma = 0.1$ for both datasets. During training, we augment the input images through a horizental flip.

The second benchmark is a Multi-Class Classification benchmark that makes use of the CIFAR-S dataset (Wang et al., 2020b). Following prior work (Wang et al., 2020b), we train a Resnet18 model for a total of 200 epochs. We train the model with Stochastic Gradient Descent (SGD) with learning rate 0.1, momentum 0.9. We use an exponential decay scheduler at epochs 50,100, and 200 with $\gamma = 0.1$. We augment the input images using a horizontal flip and a random crop.

Our method, as outlined in Section 3.2 trains a multi-class prediction head on top of the debiased feature space where the gradients are stopped from flowing back into the main network. At each epoch, we train the layer with the same hyper-parameters and total number of epochs outlined for the main model and then reset the layer at each epoch.