Diffusing DeBias: Synthetic Bias Amplification for Model Debiasing

Massimiliano Ciranni[1*] **Vito Paolo Pastore**[1,2*] **Roberto Di Via**[1*]
Enzo Tartaglione[3] **Francesca Odone**[1] **Vittorio Murino**[2,4]

[1]MaLGa-DIBRIS, University of Genoa, Italy
[2]AI for Good (AIGO), Istituto Italiano di Tecnologia, Genova, Italy
[3]Telècom-Paris, Ecole Polytechnique Superior, France
[4]Department of Computer Science, University of Verona, Italy

{massimiliano.ciranni, roberto.divia}@edu.unige.it, vito.paolo.pastore@unige.it
enzo.tartaglione@telecom-paris.fr, francesca.odone@unige.it, vittorio.murino@iit.it

Abstract

The effectiveness of deep learning models in classification tasks is often challenged by the quality and quantity of training data whenever they are affected by strong spurious correlations between specific attributes and target labels. This results in a form of bias affecting training data, which typically leads to unrecoverable weak generalization in prediction. This paper addresses this problem by leveraging bias amplification with generated synthetic data only: we introduce Diffusing DeBias (DDB), a novel approach acting as a plug-in for common methods of unsupervised model debiasing, exploiting the inherent bias-learning tendency of diffusion models in data generation. Specifically, our approach adopts conditional diffusion models to generate synthetic bias-aligned images, which fully replace the original training set for learning an effective bias amplifier model to be subsequently incorporated into an end-to-end and a two-step unsupervised debiasing approach. By tackling the fundamental issue of bias-conflicting training samples' memorization in learning auxiliary models, typical of this type of technique, our proposed method outperforms the current state-of-the-art in multiple benchmark datasets, demonstrating its potential as a versatile and effective tool for tackling bias in deep learning models. Code is available at `https://github.com/Malga-Vision/DiffusingDeBias`.

1 Introduction

Deep learning models have shown impressive results in various vision tasks, including image classification. However, their success is highly dependent on the quality and representativeness of the training data. When a deep neural network is trained on a biased dataset, so-called "shortcuts", corresponding to spurious correlations (*i.e.*, *biases*) between irrelevant attributes and labels, are learned instead of semantic class attributes, affecting model generalization and impacting test performances. In other words, the model becomes biased towards specific training sub-populations presenting biases, known as *bias-aligned*, while samples not affected by the bias are referred to as *bias-conflicting* or unbiased. In this scenario, where it is generally assumed that most of the training samples available in a biased data set are biased (typically, 95% or higher [39]), several debiasing methods have been proposed addressing the problem from different points of view [23, 46, 41, 22, 29]. When annotations on bias attributes are available, common *supervised* debiasing strategies involve reweighting the training set to give more importance to the few available bias-conflicting data [29], for example, by upsampling [31] or upweighting [43] them. Conversely, *unsupervised* debiasing methods, which

[*]These authors contributed equally to this work.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-3893

assume no prior availability of bias information, typically involve an *auxiliary model* to estimate and convey information on bias for reweighting the training set, employing its signal to guide the training of a debiased target model, while diminishing its tendency to learn bias shortcuts.

To this end, there are several existing works [39, 23, 29, 44] exploiting *intentionally biased* auxiliary models, trained to overfit bias-aligned samples, thus being very confident in both correctly predicting bias-aligned training samples as well as misclassifying bias-conflicting ones [29]. Such a distinct behavior between the two subpopulations is required to convey effective bias supervisory signals for target model debiasing, including gradients or per-sample loss values, which, *e.g.*, should ideally be as high as possible for bias-conflicting and as low as possible for bias-aligned [39]. However, such desired behavior is severely hindered by bias-conflicting training samples, despite their low number. Indeed, the low cardinality of this population impedes a suitable modeling of the correct data distribution, yet, they are sufficient to harm the estimation of a reliable bias-aligned data distribution, despite the latter having much higher cardinality. This scenario occurs under two main conditions. First, bias-conflicting samples in the training set act as noisy labels, affecting the bias learning for the auxiliary models [29]. Second, these models tend to overfit and memorize bias-conflicting training samples within very few epochs, failing to provide different learning signals for the two subpopulations [9]. Despite the available strategies, including the usage of the Generalized Cross Entropy (GCE) loss function [39, 41], ensembles of auxiliary models [23, 29], or even *ad hoc* solutions tailored to the specific dataset considered (*e.g.*, training for only one epoch or assuming the availability of a bias-annotated validation [35]), how to define a protocol for effective and robust auxiliary model training for unsupervised debiasing remains an open issue.

Remarkably, recent works [41, 29, 23] underline how the target model generalization is profoundly impacted by the protocol used to train such an auxiliary model. In principle, if one could remove all bias-conflicting samples from the training set, it would be possible to train a bias-capturing model robust to bias-conflicting samples' memorization and interference, and capable of conveying ideal learning signals [29].

To this aim, our intuition is to elude these drawbacks by skipping the use of the actual training data in favor of another set of samples synthetically generated by an adequately trained diffusion model. Specifically, we propose to circumvent bias-conflicting interference on auxiliary models leveraging *synthetic biased* data obtained by exploiting Conditional Diffusion Probabilistic Models (CDPMs). We aim to learn a *per-class biased* image distribution, from which we sample an amplified synthetic bias-aligned subpopulation. The resulting synthetic bias-aligned samples are then exploited for training a *Bias Amplifier* model, instead of using the original actual training set, solving the issue of bias-conflicting overfitting and interference by construction, as the bias-capturing model does not see the original training set *at all*. We refer to the resulting approach as *Diffusing DeBias* (DDB). The intuition is indirectly supported by recent works studying the problem of biased datasets for image generation tasks [8, 24], and showing how diffusion models' generations can be biased, as a consequence of bias present in the training sets (See Sec. 2).

This tendency can be seen as an undesirable behavior of generative models, which affects the fairness of generated images, and recent articles have started to propose solutions to decrease the inclination of diffusion models to learn biases [21, 24, 42]. However, we claim that what is considered a drawback for generative tasks can be turned into an advantage for image classification's unsupervised model debiasing, as it allows the training of a robust auxiliary model that can ideally be plugged into various debiasing schemes, solving bias-conflicting training samples memorization by construction. For this reason, we refer to DDB as a plug-in for debiasing methods in image classification. To prove its effectiveness and versatility, we incorporate DDB's bias amplifier into two different debiasing frameworks, which we refer to as *Recipe I* and *Recipe II*, including a two-step and an end-to-end approach (see Sec. 2). Specifically, we exploit our bias-amplifier to: (i) extract subpopulations further used as pseudo-labels within the popular G-DRO [43] algorithm (Recipe I, Sec. 3.3.1), and (ii) provide a loss function for the training set to be used within an end-to-end method (Recipe II, Sec. 3.3.2). Both approaches are competitive with the state-of-the-art on popular benchmark datasets. In summary, our main contributions can be regarded as follows:

- We introduce DDB, a novel and effective debiasing framework that exploits CDPMs to learn *per-class* bias-aligned distributions in biased datasets with unknown bias information. The resulting CDPM is used to generate synthetic bias-aligned images later used to train a robust bias amplifier model, thus avoiding real bias-conflicting training sample memorization and construction interference (Sec. 3).

- We propose the usage of DDB as a versatile plug-in for debiasing approaches, designing two *recipes*: one based on a two-step procedure (Sec. 3.3.1), and the other as an end-to-end (Sec. 3.3.2) algorithm. Both proposed approaches prove to be effective, achieving state-of-the-art results (Sec. 4) across popular benchmark datasets, characterized by different types of single or multiple biases.

2 Related Work

In this section, we provide a broad categorization of model debiasing works.

Supervised Debiasing. Supervised methods exploit available prior knowledge about the bias (*e.g.*, bias attribute annotations), indicating whether a sample exhibits a particular bias (or not). A common strategy consists of training a *bias classifier* to predict such attributes so that it can guide the *target* model to extract unbiased representations [1, 48]. Alternatively, bias labels can be thought of as pre-defined encoded subgroups among target classes, allowing for the exploitation of robust training techniques such as G-DRO [43]. At the same time, bias information can be employed to apply regularization schemes aiming at forcing invariance over bias features [3, 45].

Unsupervised Debiasing. It considers bias information unavailable (the most likely scenario in real-world applications), thus having the highest potential impact among the described paradigms. Among such methods, a popular strategy consists in involving auxiliary models at any level of the debiasing process. We categorize these methods into two-step [35, 44, 22], and end-to-end [39, 33], in the following paragraphs.

End-to-end approaches. Here, bias mitigation is performed online with dedicated optimization objectives. Employing techniques such as ensembling and noise-robust losses has been proposed, though they require careful tuning of all the hyperparameters, which may require the availability of validation sets with bias annotations to work. In Learning from Failure (LfF), Nam *et al.* [39] assume that bias features are learned earlier than task-related ones: they employ a *bias-capturing* model to focus on easier (bias-aligned) samples through the GCE loss [53]. At the same time, a debiased network is jointly trained to assign larger weights to samples that the bias-capturing model struggles to discriminate. In DebiAN, Li *et al.* [33] jointly train an auxiliary model (the *discoverer*) and a *classifier* (the target model) in an alternate scheme so that the discoverer can identify bias learned by the classifier during training, optimizing an ad-hoc loss function for bias mitigation.

Two-step methods. Here, the auxiliary model is employed in a first step to identify bias in training data, generally through pseudo-labeling, while the second stage consists of actual bias mitigation exploiting the inferred pseudo-labels. In [35], the auxiliary model is trained with standard ERM and then used in inference on the training set, considering misclassified samples as bias-conflicting and vice-versa: debiasing is brought on by up-sampling the predicted bias-conflicting samples. Notably, they must rely on a validation set with bias annotations to stop the auxiliary model's training after a few epochs to avoid memorization and to tune the most effective upsampling factor. In [23], a set of auxiliary classifiers is ensembled (*bootstrap ensembling*) into a *bias-commmittee*, proposing that debiasing can be performed using a weighted ERM, where the weights are proportional to the number of models in the ensemble misclassifying a certain sample. Finally, a recent trend involves the usage of vision-language models to identify bias, further exploiting bias predictions to mitigate biases in classification models [10, 25, 52].

Among the described methods, some exploit the auxiliary model for bias amplification ([29, 39, 44]) under the general assumption that a bias amplification model would either misclassify or be less confident on bias-conflicting samples. In this case, the final debiasing performance is heavily dependent on such behavior, and specifically, on *how strongly* the auxiliary model captures the bias in training data [41, 29]. However, this fundamental assumption only holds if bias-conflicting samples are not memorized during training, which is likely to happen in very few training epochs [9, 41], especially considering that the very few training bias-conflicting samples are more likely to be overfitted, as reported in [43, 5]. One could argue that an annotated validation set could be used for properly regularizing the bias amplification model, but the latter cannot be assumed available a priori in real cases, being this an emerging argument of discussion among the community [9, 41]. For this reason, differently from the previously cited research, we propose in this work that synthetic bias-aligned samples could be used *instead of* the original training set, solving the bias-conflicting

 https://doi.org/10.52202/085713-3893

memorization issue by construction. However, a real-world scenario using data with unknown bias poses a problem for generating a *pure* distribution of bias-aligned samples. As a solution, we propose to leverage diffusion models for generating a synthetic bias-aligned training set to be used for training a bias amplifier.

While the vast majority of available works regarding generative models and debiasing refer to strategies for obtaining unbiased synthetic data generation [4, 11], to the best of our knowledge, we are the first to propose the use of generative models for obtaining an amplified biased distribution.

The closest work employing generative models for model debiasing is Jung *et al.* [19], where a debiasing method exploiting a GAN [12] is introduced, which is trained for style-transfer between several *biased* appearances with different objectives. Here, a target model is debiased through contrastive learning between images affected by different biases for bias-invariant representations. Similar strategies, where bias-conflicting samples are augmented to support model debiasing, can be found in a few other works [34, 18, 28]. In [28], the problem of bias-conflicting memorization is stressed, proposing a bias-unsupervised strategy to augment and increase the diversity of bias-conflicting features, exploiting a bias amplifier analogously to [39] for their identification, while improving the generalization of a target debiased model. In [34], an intentionally biased auxiliary model is exploited, which is trained on the biased dataset, supplementing bias-conflicting samples for the target debiased model. Here, images are generated by solving an optimization problem consisting of attacking the auxiliary model predictions while preserving the ones of the target debiased model. In [18], bias-conflicting samples are again augmented exploiting the popular mix-up approach [50]. Specifically, an auxiliary model is employed to amplify easy-to-learn features, and mix-up pairs with either the same ground truth label, but different biased features (evaluated with clustering in the feature space of an intentionally biased model), or samples with different biased features (*i.e.*, belonging to different clusters), but showing the same ground truth label. Other works have also explored generative models for debiasing. For instance, BiaSwap [22] utilizes image translation techniques to swap bias attributes between images, aiming to create more balanced training data. However, methods that attempt to create synthetic bias-conflicting samples [22, 34] may require high semantic fidelity, while our method creates a functionally biased dataset, which we believe to be a simpler and more robust objective.

Differently from the previously described approaches, we are proposing an alternative strategy to avoid bias-conflicting memorization, while significantly improving the generalization of the target debiased model. In our work, we show how it is possible to leverage the intrinsic tendency of diffusion models to capture bias in data, using conditioned generations to infer the biased distribution of each target class, thus allowing the sampling of new synthetic images, sharing the same bias pattern of the original training data. Our method designs a memorization-free bias amplified model, capable of providing precise supervisory signals for accurate model debiasing, virtually compatible with any debiasing approach involving auxiliary models.

3 The Approach

Our proposed debiasing approach is schematically depicted in Figure 1. In this Section, we first describe the general problem setting (Sec. 3.1), then detail DDB's two main components: *bias diffusion* (Sec. 3.2) and the two *Recipes* for model debiasing (Sec. 3.3).

3.1 Problem Setting

Let us consider a general data distribution p_{data}, typically encompassing multiple factors of variation and classes, and to build a dataset of images with the associated labels $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ sampled from such a distribution. Let us also assume that the sampling process to obtain $\mathcal{D}$ is not uniform across latent factors of variations, *i.e.*, possible biases such as context, appearance, acquisition noise, viewpoint, etc.. In this case, data will not faithfully capture the true data distribution (p_{data}) just because of these bias factors. This phenomenon deeply affects the generalization capabilities of deep neural networks in classifying unseen examples not present the same biases. In the same way, we could consider $\mathcal{D}$ as the union between two sets, *i.e.*$\mathcal{D} = \mathcal{D}_{\text{unbiased}} \bigcup \mathcal{D}_{\text{biased}}$. Here, the elements of $\mathcal{D}_{\text{unbiased}}$ are uniformly sampled from p_{data} and, in $\mathcal{D}_{\text{biased}}$, they are instead sampled from a conditional distribution $p_{\text{data}}\left(\mathbf{x}, y \mid b\right)$, with $b \in B$ being some latent factor (bias attribute) from a set of possible attributes B, likely to be unknown or merely not annotated, in a realistic setting [24]. If $|\mathcal{D}_{\text{biased}}| \gg |\mathcal{D}_{\text{unbiased}}|$, optimizing a classification model f_θ over $\mathcal{D}$ likely results in biased predictions

and poor generalization. This is due to the strong correlation between b and y, often called *spurious correlation*, and denoted as $\rho(y, b)$, or just ρ for brevity [22, 43, 38]), which is dominating over the true target distribution semantics.

It is important to notice that data bias is a general problem, not only affecting classification tasks but also impacting several others such as data generation [8]. For instance, given a Conditional Diffusion Probabilistic Models (CDPM) modeled as a neural network $\tilde{g}_\phi(\mathbf{x} \mid y)$ (with parameters ϕ) that learns to approximate a conditional distribution $p(\mathbf{x} \mid y)$ from $\mathcal{D}$, we expect that its generations will be biased, as also stated in [8, 24]. While this is a strong downside for image-generation purposes, in this work, we claim that when $\rho(y, b)$ is very high (*e.g.*≥ 0.95, as generally assumed in model debiasing literature [39]), a CDPM predominantly learns the biased distribution of a specific class, *i.e.*, $\tilde{g}_\phi(\mathbf{x} \mid y) \approx p(\mathbf{x} \mid b)$ rather than $p(\mathbf{x} \mid y)$.

3.2 Diffusing the Bias

In the context of mitigating bias in classification models, the tendency of a CDPM to approximate the per-class biased distribution represents a key feature for training an auxiliary *bias amplified* model.

The Diffusion Process. The diffusion process progressively converts data into noise through a fixed Markov chain of T steps [15]. Given a data point $\mathbf{x}_0$, the forward process adds Gaussian noise according to a variance schedule $\{\beta_t\}_{t=1}^T$, resulting in noisy samples $\mathbf{x}_1, \ldots, \mathbf{x}_T$. This forward process can be formulated for any timestep t as: $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, where $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ with $\alpha_s = 1 - \beta_s$. The reverse process then gradually denoises a sample, reparameterizing each step to predict the noise ϵ using a model $\boldsymbol{\epsilon}_\theta$:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}, \tag{1}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\sigma_t = \sqrt{\beta_t}$.

Classifier-Free Guidance for Biased Image Generation. In cases where additional context or *conditioning* is available, such as a class label y, diffusion models can use this information to guide the reverse process, generating samples that better reflect the target attributes and semantics. Classifier-Free Guidance (CFG) [16] introduces a flexible conditioning approach, allowing the model to balance conditional and unconditional outputs without dedicated classifiers.
The CFG technique randomly omits conditioning during training (*e.g.*, with probability $p_{\text{uncond}} = 0.1$), enabling the model to learn both generation modalities. During the sampling process, a guidance scale w modulates the influence of conditioning. When $w = 0$, the model relies solely on the conditional model. As w increases ($w \geq 1$), the conditioning effect is intensified, potentially resulting in more distinct features linked to y, thereby increasing fidelity to the class while possibly reducing diversity, whereas lower values help to preserve diversity by decreasing the influence of conditioning. The guided noise prediction is given by:

$$\boldsymbol{\epsilon}_t = (1 + w)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t), \tag{2}$$

where $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y)$ is the noise prediction conditioned on class label y, and $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ is the unconditional noise prediction. This modified noise prediction replaces the standard $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ term in the reverse process formula (Equation 1). In this work, we empirically show how CDPM learns and amplifies the underlying biased distribution when trained on a biased dataset with strong spurious correlations, allowing bias-aligned image generation.

3.3 DDB: Bias Amplifier and Model Debiasing

As stated in Sec. 2, a typical unsupervised approach to model debiasing relies on an auxiliary intentionally-biased model, named here as *Bias Amplifier* (BA). This model can be exploited in either 2-step or end-to-end approaches, denoted here as *Recipe I* and *Recipe II*, respectively. This section describes the two debiasing recipes we employ in this work to demonstrate the effectiveness of our BA as a plug-in for unsupervised model debiasing.

 https://doi.org/10.52202/085713-3893

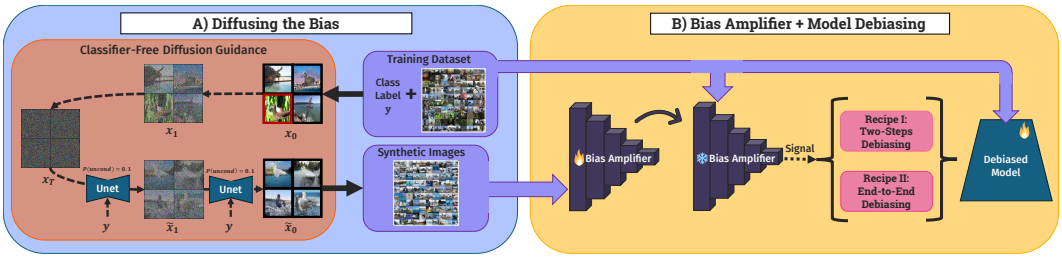
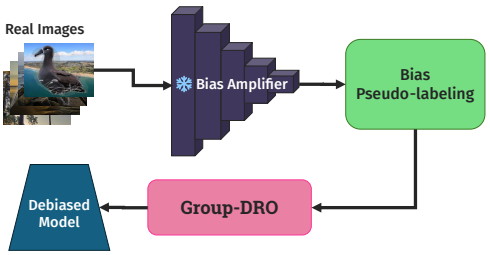
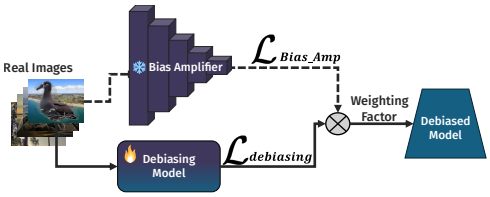


Figure 1: **Schematic representation of our DDB framework.** The debiasing process consists of two key steps: (A) *Diffusing the Bias* uses a conditional diffusion model with classifier-free guidance to generate synthetic images that preserve training dataset biases, and (B) employs a *Bias Amplifier* firstly trained on such synthetic data, and subsequently used during inference to extract supervisory bias signals from real images. These signals are used to guide the training process of a target debiased model by designing two *debiasing recipes* (*i.e.*, 2-step and end-to-end methods).



(a) Overview of *Recipe I*'s 2-step debiasing approach.



(b) Overview of *Recipe II*'s end-to-end debiasing approach.

Figure 2: Comparison of two debiasing strategies: *Recipe I* and *Recipe II*.

3.3.1 Recipe I: 2-step debiasing

The adopted 2-step approach consists of 1) applying the auxiliary model trained on biased generated data to perform a bias pseudo-labeling, hence estimating bias-aligned/bias-conflict split of original actual data, and 2) applying a *bias supervised* method to train a debiased target model for classification. For the latter, we use the group DRO algorithm [43] (G-DRO) as a proven technique for the pure debiasing step. In other words, being in the unsupervised bias scenario where the real bias labels are unknown, we estimate bias pseudo-labels performing an inference step by feeding the trained BA with the original actual training data, and identifying as bias-aligned the correctly classified samples, and as bias-conflicting those misclassified. Among possible strategies to assign bias pseudo-labels, such as feature-clustering [44] or anomaly detection [41], we adopt a simple heuristic based on the BA misclassifications. Specifically, given a sample $(\mathbf{x}_i, y_i, c_i)$ with c_i unknown pseudo-label indicating whether $\mathbf{x}_i$ is bias conflicting or aligned, we estimate bias-conflicting samples as

$$\hat{c}_i = \mathbb{1}\left(\hat{y}_i \neq y_i \ \wedge \ \mathcal{L}(\hat{y}_i, y_i) \ > \ \mu_n(\mathcal{L}) + \gamma\sigma_n(\mathcal{L})\right), \tag{3}$$

where $\mathbb{1}$ is the indicator function, $\mathcal{L}$ is the CE loss of the BA on the real sample, and μ_n and σ_n represent the average training loss and its standard deviation, respectively, depending on the loss $\mathcal{L}$. Together with the multiplier $\gamma \in \mathbb{N}$, this condition defines a sort of filter over misclassified samples, considering them as conflicting only if their loss is also higher than the mean loss increased by a quantity corresponding to a certain z-score of the per-sample training loss distribution ($\mu_n(\mathcal{L}) + \gamma\sigma_n(\mathcal{L})$ in Eq. 3). Once bias pseudo-labels over original training data are obtained, we plug in our estimate as group information for the G-DRO optimization, as schematically depicted in Figure 2a.
The above *filtering* operation refines the plain *error set*, restricting bias-conflicting sample selection to the hardest training samples, with potential benefits for the most difficult correlation settings ($\rho > 0.99$). Later in the experimental section, we provide an ablation study comparing different filtering (γ) configurations and plain error set alternatives.

3.3.2 Recipe II: end-to-end debiasing

A typical end-to-end debiasing setting includes the joint training of the target debiasing model and one [39] or more [23, 29] auxiliary intentionally-biased models. Here, we design an end-to-end

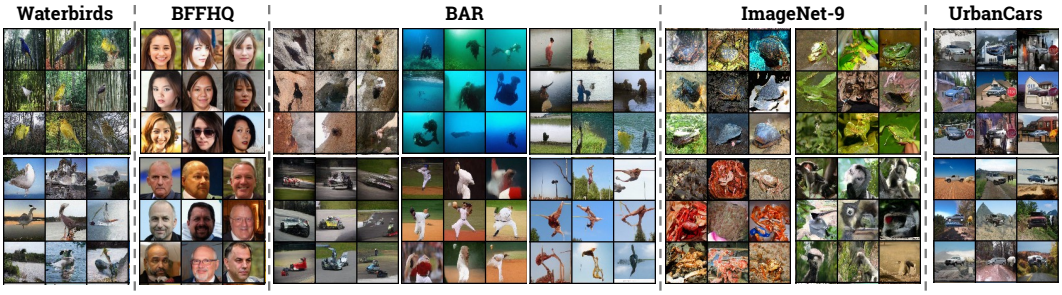


Figure 3: **Examples of synthetic bias-aligned images across multiple datasets.** Each grid shows synthetic images for specific classes across biased datasets, revealing how the model aligns with dataset-specific biases in different contexts.

debiasing procedure, denoted as *Recipe II*, incorporating our BA by customizing a widespread general scheme, introduced in the Learning from Failure (LfF) method [39]. LfF leverages an intentionally-biased model trained using Generalized CE (GCE) loss to support the simultaneous training of a debiased model adopting the CE loss re-weighted by a per-sample relative difficulty score. Specifically, we replace the GCE biased model with our Bias Amplifier, which is frozen and only employed in inference to compute its loss function for each original training sample ($\mathcal{L}_{\text{bias\_amp}}$), as schematically represented in Figure 2b. Such a loss function is used to obtain a weighting factor for the target model loss function, defined as $r = \frac{\mathcal{L}_{\text{Bias\_Amp}}}{\mathcal{L}_{\text{debiasing}} + \mathcal{L}_{\text{Bias\_Amp}}}$. Coarsely speaking, r should be low for bias-aligned and high for bias-conflicting samples.

4 Experiments

In this Section, we report the description of the benchmark datasets employed, followed by a discussion of the obtained results and a thorough ablation analysis. In the Appendix, we report the general implementation (Sec. A.2) and training details, including the implementation details of *Recipe I* (Sec. A.3) and *Recipe II* (Sec. A.4), more examples and analysis on the generated images (Sec. A.6), and additional debiasing results on less biased settings for Waterbirds plus an alternative version of BAR having known ρ and bias-annotations.

4.1 Datasets

Our experiments utilize six distinct datasets: three typical benchmark datasets, including Waterbirds [43], Biased Flickr-Faces-HQ (BFFHQ) [22], and Biased Action Recognition (BAR), taken from the original version as introduced in [39]. On top of these benchmark datasets, we are interested in evaluating our approach in challenging scenarios involving real-world images with natural biases and datasets where multiple shortcuts are present. For this reason, we include ImageNet-9 [2]/ImageNet-A [14], and UrbanCars [32] in our experiments. More dataset details are in Sec. A.1.

4.2 Results

4.2.1 Synthetic bias-aligned image generation

First, we present qualitative image generation results across multiple datasets (see Fig. 3) to validate the effectiveness of our bias diffusion approach. The generated samples maintain good fidelity and capture the typical bias patterns of each training set per class, confirming the conditioned diffusion process's ability to learn and diffuse inherent biases. This is confirmed by screening the generated images by three independent annotators, who were asked to verify (or deny) the presence of the bias in each synthetic sample, where possible. Specifically, for Waterbirds, the reported average bias-aligned proportion of synthetic generated images is 98.35%, in BAR it is 99.20%, and 99.70% is obtained for BFFHQ. All these values are higher than the known proportions of bias in the original datasets. In addition, focusing on Waterbirds, we also evaluated the generated image content by adopting an image captioner, specifically BLIP-2 [30]. We estimate a caption for each image per

https://doi.org/10.52202/085713-3893

class and count the frequency of every word. Supplementary Material (Sec. A.8) shows a histogram of the detected keywords, in which it can be noted that together with a high frequency of the object class (*bird*), similar significant bins of the bias attributes (*land*, *water*) are also evidenced.

4.2.2 Classification accuracy

Tables 1a, 1b, 2a, 2b, and 3 report the average and standard deviation of our employed metrics, obtained over three independent runs for all the considered benchmark datasets. We dedicate a separate table for each dataset, as not all methods share the same benchmarks. We report Worst Group Accuracy (WGA) for Waterbirds, as in [43, 29]. For BFFHQ, we report the average and the conflicting accuracy (*i.e.*, the average accuracy computed only on the bias-conflicting test samples), following [22, 33]. BAR and ImageNet-A do not provide bias-attribute annotations, hence, we report just the average test accuracy. For simplicity, throughout this section, we will refer to *Recipe I* and *Recipe II* as DDB-I and DDB-II, respectively. It is worth noticing that, for BAR and BFFHQ, we report benchmark results related to works that employ the same versions as ours. Across all the considered

Table 1: Results of DDB Recipes I and II on Waterbirds (left) and BAR (right). Ticks (✓) mark bias-unsupervised methods. Best unsupervised results are in bold; second-best are underlined.

(a) Debiasing results on Waterbirds

Method	Unsup	Waterbirds
		WGA
ERM	–	62.60± 0.30
LISA [49]	–	89.20
G-DRO [43]	–	91.40± 1.10
George [44]	✓	76.20± 2.00
JTT [35]	✓	83.80± 1.20
CNC [51]	✓	88.50± 0.30
B2T+G-DRO [25]	✓	90.70± 0.30
LfF [39]	✓	78.00
CDvG+LfF [19]	✓	84.80
MoDAD [41]	✓	89.43± 1.69
DDB-II (ours)	✓	**91.56**± 0.15
DDB-I (ours)	✓	90.81± 0.68
DDB-I (w/ err. set)	✓	90.34± 0.41

(b) Debiasing results on BAR

Method	Unsup	BAR
		Avg.
ERM	–	51.85± 5.92
BiaSwap [22]	✓	52.40± -
JTT [35]	✓	68.53± 3.29
LfF [39]	✓	62.98± 2.76
LWBC [23]	✓	62.03± 2.76
DebiAN [33]	✓	69.88± 2.92
MoDAD [41]	✓	69.83± 0.72
DDB-II (ours)	✓	**72.81**± 1.02
DDB-I (ours)	✓	70.40± 1.41
DDB-I (w/ err. set)	✓	70.59± 0.19

benchmark datasets, both DDB recipes show state-of-the-art results. Notably, in Waterbirds, DDB's WGA is higher than all the other state-of-the-art unsupervised methods' accuracies, as well as of that of the supervised G-DRO [43] (91.56% vs. 91.40% of G-DRO). DDB-I is providing slightly worse performance than DDB-II, but still better than the state of the art. In BAR, DDB-II is reaching

Table 2: Results of DDB Recipes I and II on BFFHQ (left) and ImageNet-9/A (right) from [22]. Ticks (✓) mark bias-unsupervised methods. Best unsupervised results are in bold; second-best are underlined.

(a) Debiasing results on BFFHQ

Method	Unsup	BFFHQ	
		Avg.	Confl.
ERM	–	–	60.13± 0.46
BiaSwap [22]	✓	–	58.87± -
JTT [35]	✓	–	62.20± 1.34
LfF [39]	✓	–	62.97± 3.22
ETF-Debias [47]	✓	–	73.60± 1.22
Park et al. [40]	✓	71.68	–
CDvG+LfF [19]	✓	–	62.20±0.45
DebiAN [33]	✓	–	62.80± 0.60
MoDAD [41]	✓	–	68.33± 2.89
DDB-II (ours)	✓	**83.15**± 1.76	70.93± 0.14
DDB-I (ours)	✓	81.27± 0.88	**74.67**± 2.37
DDB-I (w/ err. set)	✓	82.44± 0.64	71.40± 0.92

(b) Debiasing results on ImageNet-9/A

Method	Unsup	ImageNet-A
		Avg.
ERM	–	30.30
LWBC [23]	✓	35.97±0.49
CDvG+LfF [19]	✓	34.60
DDB-II (ours)	✓	37.53±0.82
DDB-I (ours)	✓	**39.80**±0.50
DDB-I (w/ err. set)	✓	38.12±0.96

https://doi.org/10.52202/085713-3893

72.81% average accuracy, which is 2.93% better than the second best (DebiAN [33]), in terms of conflicting accuracy. In BFFHQ, our DDB-I reaches an accuracy of 74.67%, with an average improvement of 1% when compared to the state-of-the-art ETF-Debias [47] (73.60%). For UrbanCars

Table 3: Comparison of deep learning methods on the UrbanCars biased dataset. ID.ACC is in-distribution accuracy (the higher, the better); other three metrics show the accuracy gap between each minority group and ID.ACC (the lower, the better). GroupDRO and LLE are separated as they rely on bias information.

Method	**ID.ACC** ↑	**BG-Gap** ↓	**CoObj-Gap** ↓	**BG-CoObj-Gap** ↓
GroupDRO [43]	91.60	10.90	3.60	16.40
LLE [32]	96.70	<u>2.10</u>	<u>2.70</u>	<u>5.90</u>
LfF [39]	<u>97.20</u>	11.60	18.40	63.20
JTT [35]	95.90	8.10	13.30	40.10
EIIL [7]	95.50	4.20	24.70	44.90
DebiAN [33]	<u>98.00</u>	14.90	10.50	69.00
DDB-I (ours)	86.39± 0.74	**1.85**± 3.21	**0.52**± 1.38	**0.12**± 1.56
DDB-II (ours)	**98.56**± 0.92	2.30± 0.60	11.10± 1.20	46.70± 2.42

(see Table 3), presenting multiple biases, we adopted the same metrics of [32] (see Suppl. Material, Sec. A.1). DDB-I shows overall the most balanced performance across bias-conflicting samples, reporting an improvement over LLE [32] of 0.25% for BG-Gap, 2.18% on CoObj-Gap, and 5.78% for BG-CoObj-Gap. DDB-II provides a significant improvement over the original LfF, but struggles more with the background-conflicting samples than DDB-I.

4.3 Ablation analysis

In this section, we provide an extensive ablation analysis of DDB main components. Without losing generality, all ablations are performed on Waterbirds and exploit Recipe I for model debiasing. Additional ablations on Recipe II can be found in the Supp. Mat. The reported analyses regard several aspects of the data generation process with respect to the Bias Amplifier, the effects of the filtering option in Recipe I, and the behavior of DDB on unbiased data.

Bias Amplifier data requirements. All our main results are obtained by training a BA on 1000 synthetic images per class, and we want here to measure how this choice impacts its effectiveness. From Table 4 (left), we observe that even 100 synthetic images are enough for good debiasing performance despite not being competitive with top accuracy scores. Increasing the number of training images improves WGA, reaching the best performance for 1000 images per class.

CFG strength and generation bias. The most important free parameter for the CDPM with CFG is the guidance conditioning *strength* w. Such strength can affect the variety and consistency of generated images. Here, we investigate DDB dependency on this parameter. Table 4 (right) shows a limited impact on debiasing performance. WGA variations along different w values equal or larger than 1 are limited, with the best performance obtained for $w = 1$.

Table 4: Ablations on: (left) synthetic image count for training the BA on Waterbirds; (right) Impact of guidance strength w on CDPM sampling, evaluated on Waterbirds. For both experiments, we employ *Recipe I*.

# Synth. images for BA training					
# Imgs.	100	200	500	1000	2000
WGA	86.25	87.10	87.67	89.05	**90.81**

Guidance strength					
w	0	**1**	2	3	5
WGA	87.85	**90.81**	89.25	89.56	88.32

Group extraction via BA plain error sets. In Sec. 3.3.1, we describe how we filter the BA *error set* through a simple heuristic based on per-sample loss distribution. In Tables 1a, 1b, 2a, and 2b, the last row (DDB-I w/ plain err. set) reports our obtained results when considering as bias-conflicting all the samples misclassified by the Bias Amplifier. Our results show a trade-off between the two alternatives. The entire set of misclassifications is always enough to reach high bias mitigation performance with less critical settings (e.g., like in Waterbirds, $\rho = 0.950$), gaining an advantage from such a coarser selection. On the other hand, the importance of a more precise selection is highlighted by the superior performance of the filtered alternative in the more challenging scenarios, like in BFFHQ ($\rho = 0.995$).

https://doi.org/10.52202/085713-3893

Bias-Identification Accuracy. To frame the goodness of our BA in correctly identifying aligned and conflicting training samples, we provide the accuracy of such a process when considering samples misclassified by it. Specifically, we reach a high bias-identification accuracy of 89.00% for Waterbirds and 96.00% on BFFHQ, supporting the effectiveness of the proposed protocol, which is not dependent on any careful regularization, bias-annotated validation sets, or large ensembles of auxiliary classifiers.

DDB impact on unbiased dataset. In real-world settings, it is generally unknown whether bias is present in datasets or not. Consequently, an effective approach, while mitigating bias dependency in the case of a biased dataset, should also not degrade the performance when the bias is not present. To evaluate DDB in this regard, we apply the entire pipeline (adopting Recipe I) on a common image dataset such as CIFAR-10 [27], comparing it with traditional ERM training with CE loss. In this controlled case, *Recipe I* provides a test accuracy of 84.16%, with a slight improvement of $\sim 1\%$ compared to traditional ERM (83.26%) employing the same target model (ResNet18). This result shows that our Bias Amplifier can still provide beneficial signals to the target model in standard scenarios, potentially favoring the learning of the most complex samples.

Effectiveness of Synthetic Data for BA Training. A core claim of our work is that training the Bias Amplifier on synthetic data prevents the memorization of real bias-conflicting samples. To validate this, we compare our standard BA (trained on synthetic data) with a BA trained on the original real training data within our two recipes. Using the real dataset significantly degrades performance. For Recipe I, the BA trained on real data for 50 epochs memorized all conflicting samples, leading to empty pseudo-label groups. Training for just one epoch (as in JTT [35]) yielded a WGA of only 79.43%. For Recipe II, using a frozen BA trained on real data resulted in a WGA of 78.45%, far below our proposed method and in line with the original LfF performance. These results empirically confirm that using synthetic data is crucial for creating an effective BA.

5 Conclusions

In this work, we present Diffusing DeBias (DDB), a novel debiasing framework capable of learning the per-class bias-aligned training data distribution, leveraging a CDPM. Synthetic images sampled from the inferred biased distribution are used to train a Bias Amplifier, employed to provide a supervisory signal for training a debiased model. The usage of synthetic bias-aligned data is empirically shown to prevent the detrimental effects typically impacting auxiliary models' training in unsupervised debiasing schemes, such as bias-conflicting sample overfitting and interference, which lead to suboptimal generalization performance of the debiased target model. Our experiments confirm that a bias amplifier trained on real data fails at this task, while our approach creates a robust and effective supervisory signal. DDB is versatile, acting as a plug-in for unsupervised debiasing methods, since its Bias Amplifier can be incorporated into several debiasing approaches. In this work, we incorporate DDB's Bias Amplifier in a two-step and an end-to-end debiasing recipes, outperforming state-of-the-art works on popular biased benchmark datasets (considering single and multiple biases), including recent works relying on vision-language models [25, 6]. Finally, while effectively mitigating bias dependency for biased datasets, DDB does not degrade performance when used with unbiased data, making it suitable to be adopted as an effective general-purpose standard training procedure.

Limitations. DDB's main limitations stem from the high computational cost of training diffusion models. As a trade-off between efficiency and generation quality, we reduce the training image resolution to 64×64; nonetheless, training our CDPM still requires $\sim$14 hours on an NVIDIA A30 GPU with 24 GB of VRAM for the largest considered dataset (BFFHQ, 19,200 images). Our ablations show, however, that this cost can be substantially reduced by training for fewer iterations or at lower resolutions (e.g., 3.2 hours for Waterbirds at 32×32) with only a minor impact on final performance. Importantly, this computational cost should be weighed against the significant, often impractical, burden of manually annotating a bias-aware validation set, which many alternative methods require [34, 43]. Another limitation inherited from diffusion models is that generation quality may decrease if the training dataset is too small, potentially limiting DDB's applicability in very small-scale settings. Finally, while we have benchmarked DDB on a wide range of standard datasets, the bias mitigation community currently lacks million-scale datasets with appropriate bias annotations for large-scale evaluation, which remains an important direction for future research.

Acknowledgements. We acknowledge the financial support from PNRR MUR Project PE0000013 "Future Artificial Intelligence Research (FAIR)", funded by the European Union – NextGenerationEU, CUP J33C24000430007, CUP C63C22000770006.

References

[1] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[2] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning (ICML)*, 2020.

[3] Carlo Alberto Barbano, Benoit Dufumier, Enzo Tartaglione, Marco Grangetto, and Pietro Gori. Unbiased supervised contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[4] Sunay Bhat, Jeffrey Jiang, Omead Pooladzandi, and Gregory Pottie. De-biasing generative models using counterfactual methods, 2023.

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[6] Massimiliano Ciranni, Luca Molinaro, Carlo Alberto Barbano, Attilio Fiandrotti, Vittorio Murino, Vito Paolo Pastore, and Enzo Tartaglione. Say my name: a model's bias discovery framework. *arXiv preprint arXiv:2408.09570*, 2024.

[7] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

[8] Moreno D'Incà, Elia Peruzzo, Massimiliano Mancini, Dejia Xu, Vidit Goel, Xingqian Xu, Zhangyang Wang, Humphrey Shi, and Nicu Sebe. Openbias: Open-set bias detection in text-to-image generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12225–12235, 2024.

[9] Mateo Espinosa Zarlenga, Swami Sankaranarayanan, Jerone TA Andrews, Zohreh Shams, Mateja Jamnik, and Alice Xiang. Efficient bias mitigation without privileged information. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024.

[10] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. *arXiv preprint arXiv:2203.14960*, 2022.

[11] Walter Gerych, Kevin Hickey, Luke Buquicchio, Kavin Chandrasekaran, Abdulaziz Alajaji, Elke A Rundensteiner, and Emmanuel Agu. Debiasing pretrained generative models by uniformly sampling semantic attributes. *Advances in Neural Information Processing Systems*, 36, 2023.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, October 2020.

[13] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[14] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022.

 https://doi.org/10.52202/085713-3893

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[18] Inwoo Hwang, Sangjun Lee, Yunhyeok Kwak, Seong Joon Oh, Damien Teney, Jin-Hwa Kim, and Byoung-Tak Zhang. Selecmix: Debiased learning by contradicting-pair sampling. *Advances in Neural Information Processing Systems*, 35:14345–14357, 2022.

[19] Yeonsung Jung, Hajin Shim, June Yong Yang, and Eunho Yang. Fighting fire with fire: Contrastive debiasing without bias-free data via generative bias-transformation, 2023.

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(12):4217–4228, 2021.

[21] Shervin Khalafi, Dongsheng Ding, and Alejandro Ribeiro. Constrained diffusion models via dual training. *CoRR*, abs/2408.15094, 2024.

[22] Eungyeup Kim, Jihyeon Lee, and Jaegul Choo. Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14992–15001, 2021.

[23] Nayeong Kim, Sehyun Hwang, Sungsoo Ahn, Jaesik Park, and Suha Kwak. Learning debiased classifier with biased committee. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 18403–18415. Curran Associates, Inc., 2022.

[24] Yeongmin Kim, Byeonghu Na, Minsang Park, JoonHo Jang, Dongjun Kim, Wanmo Kang, and Il-Chul Moon. Training unbiased diffusion models from biased dataset. In *The Twelfth International Conference on Learning Representations*, 2024.

[25] Younghyun Kim, Sangwoo Mo, Minkyu Kim, Kyungmin Lee, Jaeho Lee, and Jinwoo Shin. Discovering and mitigating visual biases through keyword explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11082–11092, 2024.

[26] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[28] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021.

[29] Jungsoo Lee, Jeonghoon Park, Daeyoung Kim, Juyoung Lee, Edward Choi, and Jaegul Choo. Revisiting the importance of amplifying bias for debiasing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(12):14974–14981, Jun. 2023.

[30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 2023.

[31] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9572–9581, 2019.

[32] Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20071–20082, 2023.

[33] Zhiheng Li, Anthony Hoogs, and Chenliang Xu. Discover and mitigate unknown biases with debiasing alternate networks. In *European Conference on Computer Vision*, pages 270–288. Springer, 2022.

[34] Jongin Lim, Youngdong Kim, Byungjai Kim, Chanho Ahn, Jinwoo Shin, Eunho Yang, and Seungju Han. Biasadv: Bias-adversarial augmentation for model debiasing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3832–3841, 2023.

[35] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR, 18–24 Jul 2021.

[36] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[38] Rémi Nahon, Van-Tam Nguyen, and Enzo Tartaglione. Mining bias-target alignment from voronoi cells. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4946–4955, 2023.

[39] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.

[40] Jeonghoon Park, Chaeyeon Chung, and Jaegul Choo. Enhancing intrinsic features for debiasing via investigating class-discerning common attributes in bias-contrastive pair. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12332–12341, 2024.

[41] Vito Paolo Pastore, Massimiliano Ciranni, Davide Marinelli, Francesca Odone, and Vittorio Murino. Looking at model debiasing through the lens of anomaly detection. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 2548–2557, February 2025.

[42] Malsha V. Perera and Vishal M. Patel. Analyzing bias in diffusion-based face generation models. In *IEEE International Joint Conference on Biometrics, IJCB 2023, Ljubljana, Slovenia, September 25-28, 2023*, pages 1–10. IEEE, 2023.

[43] Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020.

[44] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

[45] 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*, pages 13508–13517, 2021.

[46] Enzo Tartaglione, Francesca Gennari, Victor Quétu, and Marco Grangetto. Disentangling private classes through regularization. *Neurocomputing*, 554:126612, 2023.

[47] Yining Wang, Junjie Sun, Chenyue Wang, Mi Zhang, and Min Yang. Navigate beyond shortcuts: Debiased learning through the lens of neural collapse. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12322–12331, 2024.

[48] Qizhe Xie, Zihang Dai, Yulun Du, E. Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. In *NIPS*, 2017.

[49] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, pages 25407–25437. PMLR, 2022.

[50] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.

[51] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Re. Correct-n-contrast: a contrastive approach for improving robustness to spurious correlations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26484–26516. PMLR, 17–23 Jul 2022.

[52] Yuhui Zhang, Jeff Z HaoChen, Shih-Cheng Huang, Kuan-Chieh Wang, James Zou, and Serena Yeung. Diagnosing and rectifying vision models using language. *arXiv preprint arXiv:2302.04269*, 2023.

 https://doi.org/10.52202/085713-3893

[53] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems*, 2018-December:8778–8788, 5 2018.

[54] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claim of this work is proposing a model debiasing method when data bias is unknown (unsupervised bias scenario). This is pursued by preventing bias-conflicting overfitting and interference leading to suboptimal generalization performance, by leveraging the use of full synthetic data generated by a diffusion model trained on real biased data. The claims match experimental results. This is stated in Section 1 and 2.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations subsection in Section 5, including computational times.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

https://doi.org/10.52202/085713-3893

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer:[NA]

Justification: This is an application paper, and no theoretical results are included.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section A.2 reports all the experiment details necessary for replicating our work, and the source code will be released as open-source (and attached as Supplemental Material to this submission). The method is described in details in Section 3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the datasets are publicly available as open source from the sources referenced in Section A.1. Code is attached as Supplemental material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is described in Section A.1 (using the original splits in the referenced papers), and implementation details are described in Section A.2, A.3 and A.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Multiple runs are performed for each experiment, and standard deviation is reported to assess robustenss and statistical significance of our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

 https://doi.org/10.52202/085713-3893

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The hardware we used for running the experiments is described in the main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research is conducted in conform with the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: No negative impacts are expected from this research, while model debiasing can have a positive impact to reduce bias when deep learning is used for real-world applications. Such scope is reflected in the introduction.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We only use diffusion models trained from scratch on the benchmark datasets investigated in this work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the datasets and models used in this work are open source and referenced in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

 https://doi.org/10.52202/085713-3893

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code is documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human subjects, just benchmark datasets

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Only benchmark open source datasets have been used.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs have only been used for rephrasing or grammar checking.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

 https://doi.org/10.52202/085713-3893

A Technical Appendices and Supplementary Material

A.1 Datasets

- Waterbirds [43] is an image dataset exhibiting strong correlations ($\rho = 0.950$) between bird species and background environments (*e.g.*, water vs. land). It has 4,795 training images, 1,199 images for validation, and 5,794 for testing.
- BFFHQ [22] builds upon FFHQ [20] by introducing demographic biases for face recognition, while BAR [39] is crafted to challenge action recognition models that associate specific actions with particular stereotypical contexts. For these two last datasets, we specifically refer to their original versions introduced in [39] and [22], respectively: BFFHQ has a total of 19,200 training images where the semantic classes $y \in \{\text{young, old}\}$, the bias attribute $b \in \{\text{female, male}\}$, and the bias ratio $\rho = 0.995$. Then, it provides 1,000 validation images (all bias-aligned) and 1,000 testing images with uniformly distributed labels and bias attributes.
- We utilize the original BAR version as introduced in [39]. The dataset presents 6 different classes, but does not provide explicit bias annotations or a validation set, with 1,941 images for training and 654 for testing.
- ImageNet-A consists of 7,500 real-world images from a 200-class subset of ImageNet, where deep learning models consistently struggle to make correct predictions. Originally introduced by Hendrycks et al. [14] to evaluate adversarial robustness, it is also employed as a model debiasing benchmark, usually as a bias-conflicting evaluation set for models trained on ImageNet-9 [2], a collection of images with 57,200 samples where categories are super-classes of ImageNet, put together to exhibit textural and shape biases [2]. As such, both our CDPM and Bias Amplifier are trained on ImageNet-9. Then, we evaluate both recipes on ImageNet-A in terms of Average Accuracy, following the same protocol of [2, 23, 38].
- UrbanCars has been released in [32] and contains 8000 training images with the target represented by urban and country cars. The dataset is built starting from the original 196 classes of Stanford Cars [26], while synthetically incorporating two different shortcuts, background, from the Places dataset [54], and Co-Occurring Objects (CoObj), from the LVIS one [13]. In UrbanCars, 90.25% of samples present both background and CoObj attributes aligned with the target class biases, 9.5% of training samples are conflicting with respect to only one of the two attributes, while 0.25% are conflicting to both. The evaluation metrics employed for this datasets are the one introduced in [32], namely ID.ACC, BG-Gap, CoObj-Gap, BGCoObj-Gap. ID.ACC is the *in-distribution* accuracy (the weighted per-class accuracy using the training set per-group subpopulation proportions). The other metrics represent the test accuracy gap of each minority subgroup with respect to ID.ACC.

A.2 Implementation Details

In our experiments, we generate 1000 synthetic images for each target class using a CDPM implemented as a multi-scale UNet incorporating temporal and conditional embeddings for processing timestep information and class labels. For computational efficiency, input images are resized to 64×64 pixels. Input images are normalized with respect to ImageNet's mean and standard deviation. The CDPM is trained from scratch for $70,000$, $150,000$, $200,000$, 100.000, 100.000 iterations for the datasets BAR, Waterbirds, BFFHQ, ImageNet-9, and UrbanCars respectively. Batch size is set to 32, and the diffusion process is executed over $1,000$ steps with a linear noise schedule (where $\beta_1 = 1e^{-4}$ and $\beta_T = 0.028$). Optimization is performed using MSE loss and AdamW optimizer, with an initial learning rate of $1e^{-4}$, adjusted by a CosineAnnealingLR [36] scheduler with a warm-up period for the first 10% of the total iterations. As per the Bias Amplifier training, we use synthetic images from CDPM. A Densenet-121 [17] with ImageNet pre-training is employed across all datasets, featuring a single-layer linear classification head. The training uses regular Cross-Entropy loss function and the AdamW optimizer [37]. For BFFHQ, BAR, and UrbanCars, the learning rate is $1e^{-4}$, and $5e^{-4}$ for Waterbirds, and ImageNet-9. Training spans 50 epochs with weight decay $\lambda = 0.01$ except for BFFHQ, which uses $\lambda = 1.0$ for 100 epochs. We apply standard basic data augmentation strategies (following [22, 39]), including random crops and horizontal flips. For our debiasing *recipes*, we rely on the same backbones used in the existing literature, *i.e.*ResNet-18 for BFFHQ, BAR, ImageNet-9/A,

and a ResNet-50 for Waterbirds and UrbanCars [39, 22, 25, 32]. In *Recipe II*, γ (Equation 3) is set to 3 in all our main experiments. Finally, similarly to [33, 39, 41], we do not rely on a validation set for regularization. This choice is related to real-world application suitability, where no guarantees exist regarding the degree of spurious correlations present in a held-out subset of data used as validation, possibly resulting in detrimental regularization [9].

A.3 Full Implementation Details of *Recipe I* (two-step)

For the hyperparameters of *Recipe I*, we use for each dataset the following configurations. **BAR**: batch size = 32, learning rate = $5 \times e^{-5}$, weight decay = 0.01, training epochs = 80. **Waterbirds**: batch size = 128, learning rate = $5 \times e^{-4}$, weight decay = 1.0, training for 60 epochs. **BFFHQ**: batch size = 256, learning rate = $5 \times e^{-5}$, weight decay = 1.0, training for 60 epochs. The GDRO optimization is always run with the `-robust` and `-reweight_groups` flags.

For UrbanCars, presenting two biases, we apply Recipe I as it is, considering the filtered correct and wrong classifications in each class as 4 subgroups during GDRO optimization. We then consider all the available 8 subgroups at evaluation time.

A.4 Full Implementation Details of *Recipe II* (end-to-end)

For **BAR**, we use the same parameters reported in [39]: batch size = 128, learning rate = $1 \times e^{-4}$, weight decay = 0.01, training epochs = 50. **Waterbirds**: batch size = 128, learning rate = $5 \times e^{-5}$, weight decay = 0.01, training epochs = 80. **BFFHQ**: batch size = 256, learning rate = $5 \times e^{-5}$, weight decay = 0.01, training for 50 epochs. For these recipes, we accumulate gradients for 8 iterations (16 for BFFHQ) with mini-batches made of 16 samples.

A.5 Ablation Studies with DDB's *Recipe II*

This appendix section focuses on additional ablation studies, dedicated to our *Recipe II*. This set of ablations aims at providing additional evidence of Recipe II's robustness and effectiveness, similar to what is shown in Sec. 4.3 of the main paper regarding *Recipe I*. As in the main paper, we run all our ablations on Waterbirds.

Table 5: Ablation on the number of training images for the Bias Amplifier, in terms of WGA on Waterbirds, when using DDB's *Recipe II*.

# Synth. images for BA training					
# Imgs.	100	200	500	1000	2000
WGA	84.42	90.03	90.18	90.78	**91.56**

Bias amplifier data requirements. This ablation study measures how the number of training images for the Bias Amplifier impacts DDB's *Recipe II*. Results are evaluated in terms of WGA on Waterbirds (see Table 4). 200 images (100 images per class) are enough to obtain state-of-the-art competitive results (WGA equal to 90.03), with an increasing number of images corresponding to higher WGA values, up to a maximum of 91.56 for 2000 total images.

Table 6: Impact of guidance strength w on the final WGA on Waterbirds, when using DDB's *Recipe II*.

Guidance Strength					
w	0	**1**	2	3	5
WGA	87.07	**91.56**	88.01	89.87	88.30

CFG strength and generation bias. Here, we evaluate DDB's *Recipe II* dependency on the guidance strength parameter. Results are summarized in Table 6. The best results correspond to the usage of $w = 1$, with different guidance strength parameter values having a relatively limited impact on debiasing performance.

 https://doi.org/10.52202/085713-3893

DDB impact on an unbiased dataset. Similarly to what is shown in Sec. 4.3, we want to ensure that applying *Recipe II* on an unbiased dataset does not degrade the final performance. Also *Recipe II* demonstrates to be applicable in unbiased scenarios, as it measures a final test accuracy of 83.87% on CIFAR10, which is slightly higher than the ERM baseline (83.26%).

DDB with less biased datasets. We evaluated DDB's robustness on datasets with weaker spurious correlations by resampling Waterbirds to create versions with $\rho \in \{0.90, 0.80, 0.70\}$. Table 7 shows that while the ERM baseline improves as the bias weakens, DDB-I consistently provides a significant performance boost across all bias levels, demonstrating its effectiveness even when the bias is less pronounced.

Table 7: WGA (%) on Waterbirds with varying bias correlation (ρ).

Method	$\rho = 0.95$	$\rho = 0.90$	$\rho = 0.80$	$\rho = 0.70$
ERM	62.60	63.40	64.12	68.84
DDB-I	**90.81**	**86.91**	**90.19**	**86.14**

Performance on an alternative version of BAR with bias annotations. We tested out method on an alternative version of the BAR dataset, specifically the one introduced in [29] and having $\rho = 0.95$. Table 8 shows how even for this other version of BAR, our method is effective in mitigating bias dependency. When applying our *Recipe-I* we obtain SOTA performance, confirming the good results already obtained for the benchmark dataset in the main paper.

Table 8: Results on the annotated BAR dataset ($\rho = 0.95$).

Method	Average Accuracy
ERM	82.40%
ETF-Debias [46]	83.66%
DisEnt+BE [28]	84.96%
DDB-I (Ours)	**85.36% ± 0.71**

Impact of Generated Image Quality. We studied the trade-off between the visual quality of generated images, computational cost, and debiasing performance on BFFHQ. We trained the CDPM for fewer iterations, resulting in images with higher (worse) Fréchet Inception Distance (FID) scores. As shown in Table 9, performance degrades gracefully as image quality decreases. Even with a CDPM trained for only 170k iterations (FID of 30.80), the final conflicting accuracy remains competitive with the state-of-the-art and takes only $\sim$9.5 hours to train. This supports our claim that photorealistic quality is secondary to capturing the essential bias cues; even images with artifacts contain the necessary signals for our BA to function effectively.

Table 9: Impact of CDPM training iterations on FID and final conflicting accuracy (%) on BFFHQ with Recipe I.

# CDPM Iters	FID (lower is better)	Conflicting Acc.
200k (original)	22.60	74.67 ± 2.37
170k	30.80	73.70 ± 0.71
120k	36.39	71.20 ± 1.13
70k	41.82	67.40 ± 2.55

A.6 Additional Biased Generations

In this Section, we present 100 additional per class generated images for each dataset utilized in the study, as produced by our CDPM. In the Waterbirds dataset (Figure 4), the model adeptly captures and replicates the correlation between bird species and their background. In the BFFHQ dataset (Figure 8), it effectively reflects demographic biases by mirroring gender-appearance correlations. The results from the BAR dataset (Figure 7) exhibit the model's adeptness in handling multiple bias patterns, preserving typical environmental contexts for different actions. Finally, the ImageNet-9 dataset (Figure 6) stereotypes the various classes and their backgrounds well. Overall, the model

generates high-quality samples that are fidelity-biased across all datasets. Our research confirms that the model exhibits bias-learning behavior, as it not only learns but also amplifies existing biases within diverse datasets. We leverage this characteristic to produce synthetic biased data aimed at enhancing the robustness of debiasing techniques.

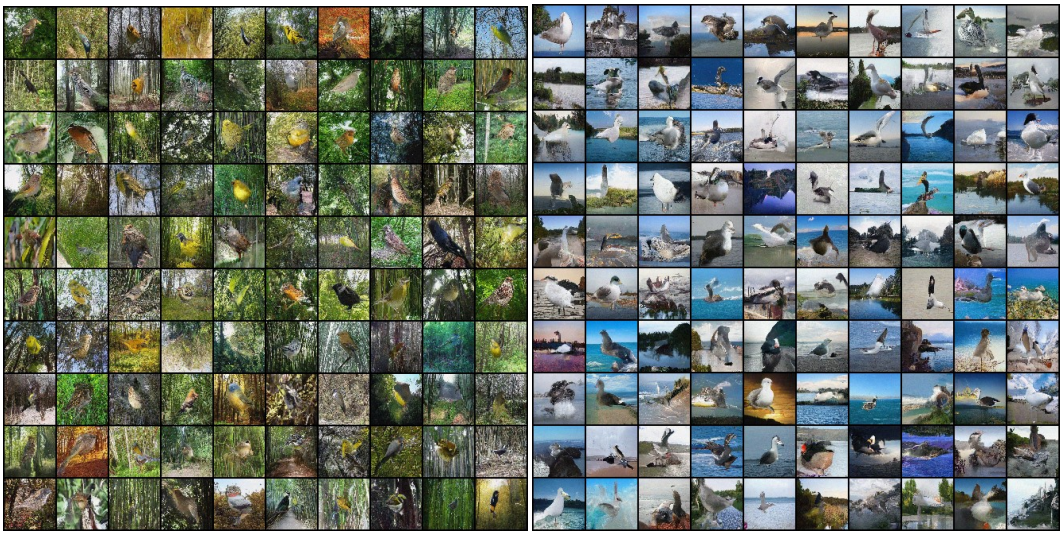


Figure 4: **Synthetic bias-aligned image generations for each class of the Waterbirds dataset.** Each grid displays 100 synthetic images per class, highlighting model bias alignment.

Figure 5: **Synthetic bias-aligned image generations for each class of the UrbanCars dataset.** Each grid displays 100 synthetic images per class, highlighting model bias alignment.

A.7 Quantitative Analysis of Bias Amplification

To empirically verify our hypothesis that a CDPM amplifies the bias present in a training dataset, we conducted a quantitative analysis on Waterbirds. We created training sets with varying degrees of spurious correlation ($\rho \in \{0.95, 0.90, 0.80, 0.70\}$) and trained a separate CDPM on each. We then generated 1,000 synthetic images per class from each trained model. These synthetic datasets were manually annotated to estimate the resulting bias correlation (ρ_{synth}).

As shown in Table 10, the CDPM consistently generated a dataset with a higher bias ratio than the one it was trained on. For example, when trained on data with $\rho = 0.95$, the generated samples

https://doi.org/10.52202/085713-3893

Figure 6: **Synthetic bias-aligned image generations for each class of the BAR dataset.** Each grid displays 100 synthetic images per class, highlighting model bias alignment.

exhibited a correlation of $\rho_{synth} = 0.99$. This empirically demonstrates the bias-amplifying property of diffusion models. This behavior can be attributed to the models learning the density of the training data and, during sampling, having a higher probability of generating samples from the high-density regions [23], which correspond to the majority bias-aligned groups in each class. This amplification is precisely what allows our BA to learn a robust representation of the bias without being exposed to the rare, real bias-conflicting samples.

Table 10: Bias amplification in synthetic data generated by a CDPM trained on Waterbirds with varying training bias ratios (ρ_{train}).

ρ_{train}	0.950	0.900	0.800	0.700
ρ_{synth}	**0.990**	**0.961**	**0.912**	**0.827**

A.8 Bias Validation via Captions in Synthetic Images

As an ulterior confirmation of bias in CDPM-generated images, besides two independent annotators, we implement an unsupervised analysis pipeline using the pre-trained BLIP-2 FlanT5$_{XL}$ model [30] for zero-shot image captioning without any task-specific prompting, for avoiding any guidance in the descriptions of the synthetic data. Our method processes the resulting captions through stop word removal and frequency analysis to reveal underlying biases without relying on predetermined categories or supervised classification models. Figure 9 presents the keyword frequency distribution for the Waterbirds dataset, where the most prevalent terms naturally correspond to known bias attributes (e.g., environmental contexts). These findings confirm that generated images are bias-aligned. Furthermore, the obtained word frequency histograms suggest that images generated by DDB's bias diffusion step may provide information useful to support bias identification (or discovery) at the dataset level.

Figure 7: **Synthetic bias-aligned image generations for each class of the ImageNet-9 dataset.** Each grid displays 100 synthetic images per class, highlighting model bias alignment.

https://doi.org/10.52202/085713-3893

Figure 8: **Synthetic bias-aligned image generations for each class of the BFFHQ dataset.** Each grid displays 100 synthetic images per class, highlighting model bias alignment.

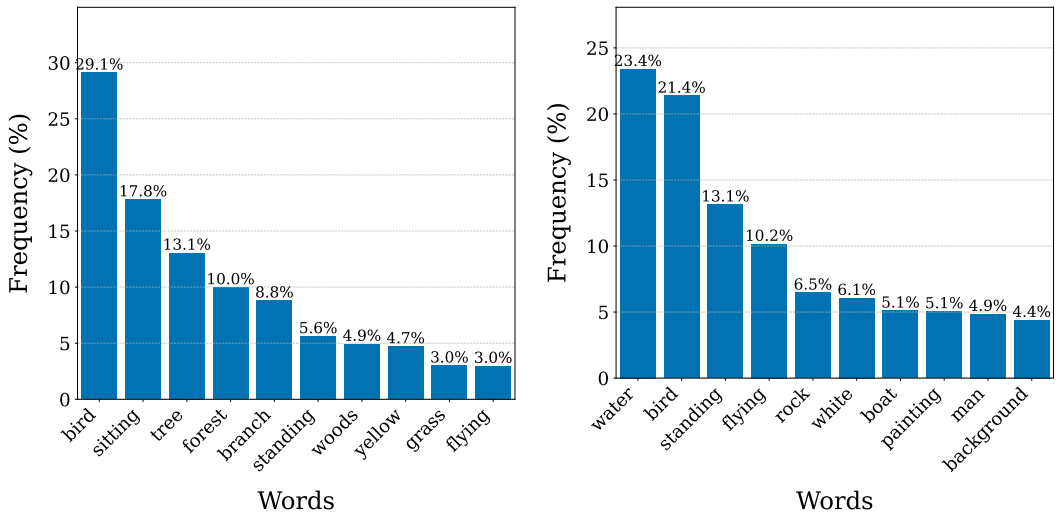


Figure 9: Word frequency analysis from 1000 generated captions for Waterbirds classes ‘landbird’ (left) and ‘waterbird’ (right), showing top 10 most frequent terms.