AracNet: Revealing Debiasing Signals across Layers with Shallow Monitors

Vito Paolo Pastore[1,2]*, Massimiliano Ciranni[1]*, Enzo Tartaglione[3], and Vittorio Murino[2]

[1] MaLGa-DIBRIS, University of Genoa, Genoa, Italy
vito.paolo.pastore@unige.it, massimiliano.ciranni@edu.unige.it
[2] Istituto Italiano di Tecnologia, Genoa, Italy
vittorio.murino@iit.it
[3] LTCI, Télécom Paris Institut Polytechnique de Paris, Paris, France
enzo.tartaglione@telecom-paris.fr

Abstract. Deep neural networks often show poor generalization when trained on biased datasets presenting spurious relations between samples and target labels. Unsupervised debiasing methods aim at mitigating models' dependency on bias, without relying on bias information. Bias shortcuts are typically learned early in training, with the model rapidly overfitting the few samples not sharing the same related attributes. Avoiding bias conflicting memorization for mining bias information useful for mitigation (e.g., in the form of pseudo-labels) is still a fundamental open issue in this context, with existing approaches proposing to learn for a few epochs or to rely on unbiased validation sets for early stopping. In this work, we propose to exploit intermediate layers' features for achieving bias mitigation robust to the memorization problem, replacing the *when to stop* with a *where to look* paradigm. Specifically, we propose AracNet, an unsupervised debiasing framework for self-mitigation. Attaching one linear layer to each network's block, which we refer to as Shallow Monitor, we obtain a useful debiasing signal capable of mitigating bias dependency in the same model when randomly reinitialized. Our results on typical benchmark datasets with single and multiple biases confirm the effectiveness of the proposed approach, paving the way for further research in understanding how bias propagates in trained models. Code is available at https://github.com/Malga-Vision/AracNet

Keywords: Model Debiasing · Image Classification

1 Introduction

Deep neural networks are sensitive to bias in training data. When training images show bias attributes highly correlated with target labels, deep models generally learn bias-related shortcuts, with poor generalization to test samples that do not possess the same bias [30]. The majority of the training samples showing bias

* Equally contributing first authors.

attributes correlated with their target class are referred to as *bias-aligned*, while we name the few remaining samples (with no or different bias attributes) as *bias-conflicting*. In the last few years, several model debiasing methods have been proposed to deal with this issue, aiming to increase models' generality, diminishing their dependency on bias-related features when making predictions [10]. Among them, supervised methods exploit bias labels to either up-weight or up-sample the few bias-conflicting samples [9], employ robust optimization techniques based on pre-defined bias groups [18,21,33], or adversarial optimization objectives [14]. However, biased information is generally not available in real-world scenarios, paving the way for the development of bias-unsupervised methods, which do not exploit prior knowledge for debiasing. In the unsupervised frameworks, existing approaches either rely on intentionally biased external models to provide indirect debiasing signals (for instance coming from individual loss values [30] or gradients [1]), or to extract bias pseudo-labels [4, 25, 31, 36] later employed for up-sample or up-weighting estimated bias-conflicting population [25, 31] or as group estimation for applying robust optimization techniques [6, 34, 36].

A foundational assumption in unsupervised methods is that neural networks naturally learn simple, bias-aligned features early in the training process before memorizing more complex, bias-conflicting examples [30]. Consequently, several methods in this context employ dedicated heuristics to determine an optimal moment to stop the training of the auxiliary intentionally biased model, as this is fundamental for optimal bias-mining [5, 8, 25, 28, 31]. Importantly, mining bias from a model that is currently underfitting or overfitting the training set (up to complete memorization) impedes a correct modeling of the actual training subpopulations [8, 20, 31], as in both cases the auxiliary model tends to provide similar learning signals for both bias-aligned and bias-conflicting samples, becoming uninformative. To tackle this crucial aspect, some methods rely on bias-annotated validation sets or train the auxiliary model (or an ensemble) for very few epochs [20, 25]. However, how to make the bias mining step robust to conflicting memorization is still an open issue. Recently, a solution to this issue has been proposed in [5]: synthetic bias-aligned samples are generated through a conditional diffusion model, later used to train an auxiliary biased model robust to bias-conflicting memorization by construction. However, it requires the training of a generative model, which brings a considerable computational overhead, requiring several compute hours to provide the synthetic dataset for the auxiliary model training.

In this context, we argue that rather than focusing on *when to stop* for effective bias mining, relying on heuristics or annotated validation sets, it is possible to shift to the *where to look* in the model as a paradigm for producing useful debiasing signals. Our intuition is that memorization predominantly happens in the last layers of the network, depending on the specific type of bias. For instance, if the bias is in the background, this may be mostly encoded in the earliest layers, while for semantic biases (such as gender in face images) this may happen in later stages of the network. Accordingly, we propose AracNet, an unsupervised self-mitigation framework based on shallow classifiers (named monitors) that

we attach to each network's block, acting as bias sensors. After a model has been pre-trained on the biased training set, it is frozen, and the monitors are trained on the intermediate feature representations coming from the corresponding model's blocks on the same classification task. As the model representation is biased, we expect corresponding monitors to be very confident on bias-aligned samples, as they rely on bias-related shortcuts encoded in their layers' features, with an opposite behavior for the few bias-conflicting samples. Leveraging this, AracNet automatically identifies the most biased layer with a bimodality score (see Section 3) computed on top of the corresponding monitor's output probability. This monitor is then exploited to provide a per-sample reweighing coefficient, constructed from its confidence and its loss, employed in a self-mitigation framework to debias the same model reinitialized with random weights. To the best of our knowledge, *this is the first work proposing to exploit signals coming from internal layers for model debiasing*, serving as a computationally light potential solution to the conflicting memorization problem, without relying on external annotated validation sets or heuristic procedures. AracNet, in fact, introduces an almost negligible computational overhead, limited to the training of a few additional linear classifiers. Results on several benchmark datasets show the effectiveness of the proposed approach, leading to competitive or state-of-the-art performance on different types of bias.

To summarize our main contributions include: (i) a self-mitigation method based on intermediate layer signals via shallow monitors, capable of automatically adapting for debiasing different sources of bias; (ii) the proposal of a potential solution to the training conflicting memorization for unsupervised debiasing methods, without the requirement of annotated validation sets or heuristic early stopping for bias mining, (iii) differently from other methods tackling the same challenge, our solution does not require computationally heavy training or pre-processing, relying only on intermediate feature representations, which as a byproduct, provide insights on how bias propagates across the network and potential indirect information on bias nature.

2 Related work

In this section, we describe the most relevant literature addressing model debiasing, focusing on unsupervised debiasing methods.

Model Debiasing and Spurious Correlations. Deep neural networks trained via standard Empirical Risk Minimization (ERM) are highly vulnerable to learning spurious correlations, often adopting shortcuts associated with bias attributes rather than the true semantic features of a target class [2, 10]. In the context of image classification, this technical bias manifests as systematic errors where the model over-relies on specific shared features (e.g., backgrounds, textures, or co-occurring objects), leading to unfair advantages or disadvantages for certain subgroups. Supervised debiasing methods address this by utilizing explicit bias

annotations to balance the learning process, often through group distributionally robust optimization [21,27,33], tailored data augmentation [22], or strategic sample reweighting [9]. However, explicit bias annotations are rarely available in real-world scenarios, making these approaches difficult to scale and driving the need for unsupervised alternatives.

Unsupervised Debiasing. Unsupervised debiasing frameworks aim to simultaneously identify and mitigate bias without relying on predefined bias labels. Many works rely on the *learned early* phenomenon regarding bias, which posits that neural networks naturally learn simple, bias-aligned features early in the training process before memorizing more complex, bias-conflicting examples [30, 33]. Prominent methods like Learning from Failure (LfF) and Just Train Twice (JTT) exploit this dynamic. JTT, for example, deliberately trains an auxiliary model for only a few epochs to identify bias-conflicting samples via misclassification, subsequently upweighting them in a second training phase [25]. Other approaches utilize a Generalized Cross Entropy (GCE) loss [42] to amplify the predictions of an early-stage, bias-capturing model [15, 20, 36].

However, a critical bottleneck in these methods is determining the optimal moment to stop the auxiliary model's training. Stopping too early results in uninformative underfitting, while stopping too late leads to the complete memorization of bias-conflicting samples [20]. Recent advancements have challenged the strict *learned early* assumption, demonstrating that models adapt to biases dynamically [29]. Alternatively, recent works have resorted to generative models, either to synthesize bias-aligned data for training the auxiliary model [6], or to apply various style transfers to training images [13]. The usage of generative models introduces significant computational overhead and complexity. An alternative line of work explored contrastive learning objectives to enforce more compact representations while retaining good average performance [12, 39], or even adversarial techniques to penalize bias reliance [24, 32]. Furthermore, approaches like [5, 16, 41] leverage foundation vision-language models to discover potential biases in an image classification model, providing natural language descriptions.

Debiasing Image Classification through Dedicated Architectures. While many debiasing strategies prioritize loss reweighting or sample clustering, an emerging line of research tackles bias mitigation directly through dedicated network architectures and internal representation analysis [35,37]. These architecture-based approaches hypothesize that structural inductive biases can inherently prevent the assimilation of spurious correlations. For instance, multi-exit architectures like OccamNets favor the simplest correct hypothesis, utilizing early exit gates to minimize the network's reliance on overly complex bias features [35]. Other frameworks, such as SIFER, introduce auxiliary layers at intermediate depths to explicitly identify and *sieve out* simple predictive features, applying a targeted forgetting loss that forces deeper layers to learn more robust, unbiased representations [37]. Furthermore, methods utilizing network pruning

demonstrate that over-parameterized models inherently contain unbiased sub-networks, which can be isolated by applying gating parameters and privacy heads to bottleneck layers [28].

AracNet as a Self-Mitigating Framework. Following past observations on how intermediate representations can provide insights into a model's reaction to spurious correlations, we propose AracNet as an effective, self-contained mitigation framework. Unlike existing unsupervised methods that struggle with the early-stopping dilemma, rely on computationally expensive generative models, or require external foundation models for open-set bias discovery, AracNet shifts the paradigm from *when to stop* to *where to look*. By leveraging feature responses across network blocks through dedicated shallow MLPs (Shallow Monitors), we demonstrate how it is possible to derive strong debiasing signals and map how bias propagates across the backbone. Because these Shallow Monitors evaluate frozen, pre-trained representations, they are robust to typical training set memorization. This allows them to provide a reliable, stable debiasing signal throughout the entire mitigation process. By automatically locating the layer where bias is most separable, AracNet seamlessly integrates bias identification and mitigation into a single, efficient pipeline. A schematic representation is provided in 1.

3 Method

3.1 Problem formulation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be a training dataset, where $x_i \in \mathcal{X} \subset \mathbb{R}^d$ being the input and $y_i \in \mathcal{Y} = \{1, \ldots, C\}$ are the target outputs.

If the dataset is biased, we assume the existence of some bias attributes $b \in \mathcal{B}$, statistically correlated with the labels in the training distribution P_{tr}, so that the mutual information $I_{P_{\text{tr}}}(y\ ;\ b) > 0$, with weaker or absent correlations in the test distribution.

Generally, the training distribution can be represented as a mixture of bias-aligned $P_A(x, y, b)$ and bias-conflicting $P_C(x, y, b)$ subpopulations

$$P_{\text{tr}}(x, y, b) = \rho\, P_A(x, y, b) + (1 - \rho)\, P_C(x, y, b), \quad \rho \in (0, 1), \tag{1}$$

where ρ represents the degree of bias-target relation, with standard benchmark datasets typically showing a $\rho \in [0.950, 0.995]$.

3.2 Representation Learning through the network layers

Let $f_\theta : \mathcal{X} \to \mathcal{Y}$ be a deep neural network trained via empirical risk minimization:

$$\min_{\theta}\ \mathbb{E}_{(x,y)\sim P_{\text{tr}}}\left[\mathcal{L}(f_\theta(x), y)\right]. \tag{2}$$

We consider f_θ decomposed into L sequential blocks (e.g., the residual blocks in a ResNet) and denote $h_\ell(x) \in \mathbb{R}^{d_\ell}$ the intermediate representation at block ℓ.

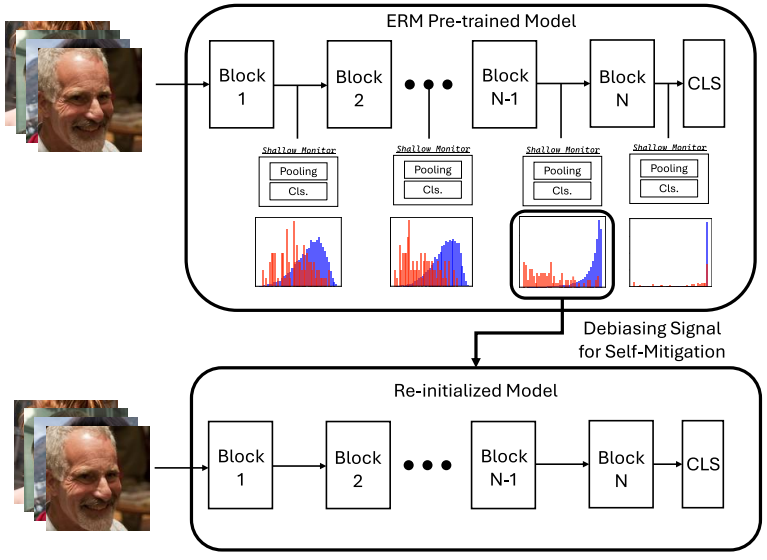

Fig. 1: Schematic overview of AracNet. The features at each network's block are given as input to a monitor, further used to provide effective debiasing signals via loss reweighting.

In [30], it is empirically shown how bias features are typically learned early in training, while in a few epochs bias-conflicting samples are memorized, with the model correctly classifying with high confidence all training samples, while still failing to generalize, as it relies on bias-related shortcuts.

In AracNet, we hypothesize that within the L blocks, there are intermediate layers where features $h_\ell(x)$ allow for a good linear separation of bias-aligned and bias-conflicting samples. As such, after training f_θ, we freeze its parameters and attach to each block ℓ a linear classifier, which we refer to as monitor.

3.3 Monitors as bias linear probes

We refer to each monitor as $m_\ell(x)$, and it corresponds to a pooling plus a single dense layer with a softmax ($m_\ell(x) = \text{softmax}(U_\ell\, h_\ell(x))$. The monitor parameters U_ℓ are updated with conventional ERM, employing a cross-entropy loss:

$$\min_{U_\ell} \; \mathbb{E}_{(x,y)\sim P_{\text{tr}}} \left[\mathcal{L}\big(m_\ell(x), y\big) \right], \tag{3}$$

while keeping the backbone f_θ frozen.

Let $c_\ell(x) = m_\ell(x)_y$ denote the confidence of monitor ℓ on its ground truth class y. We are interested in identifying the block ℓ^* with the most biased features $h^*_\ell(x)$, meaning that bias-conflicting and bias-aligned samples should be as separable as possible with these features, among the network blocks. Consequently, we expect $c^*_\ell(x)$ to be very small for bias-conflicting and very high

for bias-aligned samples. Considering that the latter are the majority of training samples in a biased dataset, when bias features dominate the representation learned at layer ℓ, we expect the confidence distribution $c_\ell(x)$ to become bimodal, with a large mode with high-confidence values for bias-aligned samples, and a tail with low confidence for bias-conflicting samples.
As such, we define a bimodality score $B(\ell)$ (see Section 4.2) computed exploiting the empirical distribution of $c_\ell(x)$ and select:

$$\ell^* = \arg\max_{\ell} B(\ell) \tag{4}$$

Importantly, this selection does not require early stopping or unbiased validation data.

3.4 Self-mitigation with monitor debiasing signals

At this stage, while maintaining a copy of the pre-trained weights for training our monitors, we randomly re-initialize f_θ to learn a target debiased model, exploiting the monitor to provide an effective debiasing signal, in a self-mitigation framework. We re-initialize the model as the pre-trained features for f_θ are severely biased, making its mitigation impractical (as shown in [25, 31]). We train the target debiased model f_θ^{deb} employing the monitor ℓ^* as a bias sensor to provide an indirect bias information incorporated in a per-sample weight for the loss function. Let $\mathrm{CE}\big(m_\ell^*(x), y\big)$ be the loss function corresponding to monitor ℓ^*, while $\mathrm{CE}\big(f_\theta^{deb}(x), y\big)$ be the loss for the target debiasing model. Our debiasing model is trained employing the following loss function:

$$\mathcal{L}(x, y; f_\theta^{deb}) = w_{\text{surprisal}} \cdot w_{\ell^*} \cdot \mathrm{CE}\big(f_\theta^{deb}(x), y\big) \tag{5}$$

$$w_{\text{surprisal}} = -\log\big(c_\ell^*(x)\big)^\lambda, \quad w_{\ell^*} = \frac{\mathrm{CE}\big(m_\ell^*(x), y\big)}{\mathrm{CE}\big(f_\theta^{deb}(x), y\big) + \mathrm{CE}\big(m_\ell^*(x), y\big)} \tag{6}$$

Here, $w_{\text{surprisal}}$ is a coefficient exploiting the surprisal for monitor ℓ^*, being high for bias-conflicting and low for bias-aligned samples, with λ being a tunable parameter that allows to further increase the weight for low-confidence and decrease the weight for high-confidence samples, while the second term is inspired by [30], being close to 0 for bias-aligned and towards 1 for bias-conflicting samples. The monitor ℓ^* and the target debiasing model are trained together, in an alternate scheme. At each epoch, we first update the monitor on the frozen features $h_{\ell^*}(x)$, and then we update the target debiasing model employing the loss described in Eq. 5, with the monitor ℓ^* kept frozen. A pseudocode description for the entire AracNet debiasing pipeline is provided in Algorithm 1.

4 Experiments

In this section, we describe the datasets used in this work, provide details on the experimental protocol, and present and discuss the obtained results.

Algorithm 1 Pseudocode for AracNet

Require: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$; backbone $f(\cdot)$ with L blocks; hyperparameter λ; constant ϵ; warmup epoch T_w

Train biased backbone: $f_{\theta^{\text{pre}}} \leftarrow \arg\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\text{CE}(f_\theta(x), y)\big]$.
Freeze backbone $f_{\theta^{\text{pre}}}$.
Initialize monitors $\{U_\ell\}_{\ell=1}^{L}$.
for $t = 1$ **to** T_w **do**
  (A) Monitor update on $f_{\theta^{\text{pre}}}$
  for $\ell = 1$ **to** L **do**
    Extract features $h_\ell(x)$ from $f_{\theta^{pre}}$.
    Compute monitor prediction $m_\ell(x) = \text{softmax}(U_\ell h_\ell(x))$.
    Update U_ℓ with SGD and $\text{CE}(m_\ell(x), y)$.
  end for
end for
(B) Select layer that maximizes bimodality
Compute confidence $c_\ell(x) = m_\ell(x)_y$.
Compute bimodality score $B(\ell)$ from the empirical distribution of $\{c_\ell(x)\}$.
Select $\ell^* \leftarrow \arg\max_\ell B(\ell)$.
Discard monitors $\{U_\ell\}_{\ell\neq\ell^*}$, keeping only U_{ℓ^*}.
for $t = T_w$ **to** T **do**
  (C) Debias model update using selected monitor
  Freeze monitor U_{ℓ^*}.
  Compute debias loss $\text{CE}(f_{\theta^{\text{deb}}}(x), y)$.
  Compute monitor loss $\text{CE}(m_{\ell^*}(x), y)$.
  Compute confidence $c = m_{\ell^*}(x)_y$.
  Compute surprisal weight $w_{\text{surprisal}} = (-\log c)^\lambda$.
  Compute monitor weight w_{ℓ^*} as per Eq. 6
  Update $f_{\theta^{\text{deb}}}$ using AdamW on $\mathcal{L}(x, y; f_\theta^{\text{deb}}) = w_s \cdot w_{\ell^*} \cdot \text{CE}(f_{\theta^{\text{deb}}}(x), y)$.
  (D) Update monitor U_{ℓ^*} **on** $f_{\theta^{\text{pre}}}$
end for
return Debiased parameters θ^{deb}.

4.1 Datasets and evaluation metrics

We employ both synthetic and realistic datasets for our evaluation.

Corrupted CIFAR-10. Introduced in [30], this dataset is built starting from the regular CIFAR-10 dataset, exploiting the image corruptions described in [11] as bias attributes. The dataset is available with four different degrees of correlation $\rho \in [0.995, 0.990, 0.980, 0.950]$. The unbiased test set presents a ratio of 90/10 for bias-conflicting and bias-aligned samples. In our results, we report the average accuracy on the test set, as done in [19, 30].

BFFHQ. Gender-Biased Flickr-Faces-HQ (BFFHQ) is introduced in [15] and presents 19.200 images with two classes: "young" and "old". Here, the bias-attribute is the perceived gender, with "young females" and "old males" being bias-aligned samples. The degree of correlation is $\rho = 0.995$, and the unbiased test set presents 50% of bias-aligned and bias-conflicting samples. In our results,

we report the accuracy on the conflicting samples, as done in [6, 15, 38].

Waterbirds. Introduced in [33], this dataset presents waterbirds and landbirds as target classes, with the bias-attribute represented by the background, such that around 95% of the training samples are either waterbirds on water background or landbirds on land background. The dataset is strongly imbalanced, with 4.795 training images and landbird training samples being 2.67 times the number of waterbird samples. The unbiased test set presents 50% of bias-aligned and bias-conflicting samples, while keeping the same class-imbalance ratio as the training set. In our results, we report the Worst Group Accuracy (WGA), as done in [16, 33, 43].

BAR. Originally introduced in [30] without explicit bias annotations, this dataset presents six classes corresponding to actions including: climbing, diving, fishing, pole vaulting, racing and throwing, with the bias represented by the scene context associated with each action. In this work, we utilize the version released in [20], providing two different degrees of correlations $\rho \in [0.990, 0.950]$, with the unbiased test set presenting only bias-conflicting samples. Here, we report the average test accuracy, as done in [32, 38].

UrbanCars. UrbanCars is a semi-realistic image classification dataset introduced in [22], where the task is to recognize between car vehicles typical of urban environments (mainly sedans) or countryside environments (e.g., off-roads, pick-ups). This dataset provides two distinct sources of bias, one coming from background correlations and the other due to co-occurring objects. Multiple bias benchmark are becoming more and more used in the literature, as it has been shown how methods usually targeting single biases may exacerbate reliance on one while mitigating dependency from the other [22, 23]. The total number of images for this benchmark is 8,000: among them, 90.25% of training samples are aligned with respect to both bias attributes, 9.50% conflicts for a single attribute, and 0.25% conflicts with both bias attributes. Performance on this dataset is measured through In-Distribution accuracy (I.D. Acc) and the gaps between accuracy on the conflicting subgroups and the I.D. Acc itself [22].

4.2 Implementation details

We adopt the same experimental protocol described in [15, 30, 31, 38]. Specifically, for all the datasets, we resize images to 224×224, but for Corrupted CIFAR-10, we keep the original resolution 32×32. We employ an ImageNet-pretrained ResNet50 backbone for Waterbirds and UrbanCars, an ImageNet-pretrained ResNet18 for BAR and BFFHQ, and a ResNet20 with random initialization for Corrupted-CIFAR10. The number of warmup epochs is fixed to 30 for all the datasets, but Corrupted-CIFAR10, where we employ 50 warmup epochs, as the model starts with random weights. For all the datasets, regardless the degree of correlation, the bimodality score $B(\ell)$ is defined starting from $c_\ell(x)$ computed with respect to the class label (see Algorithm 1), defining a

Method	Corrupted CIFAR-10			
	$\rho = 0.995$	$\rho = 0.990$	$\rho = 0.980$	$\rho = 0.950$
ERM+CE	26.59 ± 0.82	29.84 ± 2.24	34.95 ± 0.20	44.88 ± 0.08
BiaSwap [15]	29.11	32.54	35.25	41.62
LfF [30]	28.81 ± 0.44	33.07 ± 0.77	40.66 ± 0.70	50.72 ± 1.31
DFA [19]	29.95 ± 0.71	36.49 ± 1.79	41.78 ± 2.29	51.13 ± 1.28
MoDAD [31]	27.26 ± 0.47	–	41.27 ± 0.26	50.48 ± 0.42
DeNetDM [38]	$\mathbf{38.93 \pm 1.16}$	$\mathbf{44.20 \pm 0.77}$	$\underline{47.35 \pm 0.70}$	$\underline{56.30 \pm 0.42}$
AracNet (Ours)	$\underline{33.23 \pm 1.93}$	$\underline{39.61 \pm 0.28}$	$\mathbf{47.48 \pm 1.89}$	$\mathbf{56.80 \pm 0.78}$

Table 1: Results comparison on Corrupted CIFAR-10. We highlight in bold the best results from unsupervised debiasing works, second-best are underlined.

margin as the difference between the 99th percentile and the 1st percentile, further multiplied for the 50th percentile in $c_\ell(x)$, to penalize large margins arising from uni-modal (e.g., gaussian) distribution in the confidence space rather than bimodality. Batch size is fixed to 256 for Corrupted-CIFAR10, 64 for all the remaining datasets. The backbone is pre-trained with SGD and a learning rate of 0.001 for all the datasets, and the monitors are trained with SGD and a learning rate of 0.05 for all the datasets, but UrbanCars and Waterbirds, where we employ 0.005. For debiasing, we employ AdamW for all the datasets, with a learning rate of 0.005 for Corrupted-CIFAR10, $1e^{-5}$ for UrbanCars, and $5e^{-5}$ for Waterbirds, BFFHQ, and BAR. All the results report the average and standard deviation computed across three independent runs of AracNet.

4.3 Results

We evaluate AracNet across several benchmark datasets, measuring the effectiveness of the proposed method with different sources of bias (including synthetic, background, contextual, and multiple), across varying levels of bias severity.

Corrupted CIFAR-10. As shown in Table 1 AracNet consistently achieves highly competitive results across varying degrees of bias correlation (ρ). Notably, it outperforms or closely matches other unsupervised methods for lower correlation thresholds, providing the second-best results on higher degrees of correlation. This experiment confirms how AracNet's monitors can effectively capture synthetic bias.

BFFHQ. Table 3 presents the results on the gender-biased BFFHQ dataset, which presents a severe bias correlation ($\rho = 0.995$). Here, AracNet provides the best performance, with an average increasing if 0.23% over DeNetDM and of 1.26% over DDB. These results highlight how our monitors can provide effective debiasing signals even with highly biased training sets and with finer semantic biases, such as perceiving gender in face images.

Waterbirds. As shown in Table 2, AracNet achieves a WGA of 91.27 ± 0.56, establishing the highest performance compared to strong recent baselines like

Method	Waterbirds (WGA)
ERM	62.60 ± 0.30
ERM (ViT-S/16)	67.69 ± 2.14
George [36]	76.20 ± 2.00
JTT [25]	83.80 ± 1.20
CNC [40]	88.50 ± 0.30
B2T+GDRO [17]	90.70 ± 0.30
LfF [30]	78.00
SaMyNa [5]	90.62 ± 0.10
DDB [6]	$\underline{90.81 \pm 0.49}$
GERNE [3]	89.88 ± 0.67
CCDB [43]	90.48 ± 0.28
FG-CCDB [44]	90.56 ± 0.24
AracNet (ours)	$\mathbf{91.27 \pm 0.56}$
AracNet (ViT-S/16)	90.66 ± 0.49

Table 2: Results comparison on Waterbirds. We highlight in bold the best results from unsupervised debiasing works; the second-best are underlined.

Method	BFFHQ
	$\rho = 0.995$
Vanilla ERM	60.13 ± 0.46
LfF [30]	65.19
CDvG+LfF [13]	62.20 ± 0.45
DebiAN [23]	67.36
MoDAD [31]	68.33 ± 2.89
LoYS [32]	68.50 ± 0.70
DeNetDM [38]	$\underline{75.70}$
DDB [6]	74.67 ± 2.37
AracNet (ours)	$\mathbf{75.93 \pm 0.76}$

Table 3: Results comparison on BFFHQ. We highlight in bold the best results from unsupervised debiasing works; the second-best are underlined.

Method	BAR	
	$\rho = 0.990$	$\rho = 0.950$
ERM+CE	70.16	82.95
LfF [30]	70.55	82.53
AmpliBias [19]	73.30	84.67
A^2 [26]	71.15	83.07
LfF + BE [31]	73.36	83.87
DeNetDM [38]	73.84 ± 2.56	79.61 ± 3.18
DisEnt + BE [38]	73.29	84.96
LoYS [32]	$\underline{75.92 \pm 0.93}$	$\mathbf{85.00 \pm 0.66}$
AracNet (Ours)	$\mathbf{76.22 \pm 1.47}$	$\mathbf{85.00 \pm 0.73}$

Table 4: Results comparison on BAR. We highlight in bold the best results from unsupervised debiasing works; the second-best are underlined.

[3, 5, 6, 16, 43, 44]. Specifically, we improve of $\sim 0.5\%$ over methods employing vision-language foundation models for debiasing [5, 16], with a similar increase for generative-based methods [6] and unsupervised approaches based on group robustness objectives [43, 44].

BAR. On the BAR dataset (Table 4), our method shows the best overall performance, yielding 76.65 ± 1.47 for $\rho = 0.990$, with an average increase of $\sim 0.70\%$ over the second-best performing LoYS and 85.00 ± 0.73 for $\rho = 0.950$, with similar performance to LoYS and DisEnt + BE. These results confirm the effectiveness of AracNet on context bias across two different bias strength levels.

UrbanCars. Regarding the challenge provided by the presence of multiple sources of bias, in Table 5 we provide the results obtained by AracNet and a

comparison with recent SOTA methods. Similar to DDB, while AracNet shows an I.D. ACC below 90%, the bias mitigation is evident as all the gaps are very low (a negative gap in Table 5 refers to an increase over ID.ACC), with the subgroup corresponding to conflicting co-occurrent objects showing the best performance. This indicates that, at the cost of a reasonable trade-off in terms of overall accuracy, the resulting model behaves consistently across subgroups, without promoting or demoting any of them in terms of final accuracy. AracNet is capable of simultaneously mitigating both bias attributes, while methods such as EIIL, JTT, LfF end up mitigating one bias attribute while amplifying the dependence on the other (see [22]). Furthermore, AracNet provides better mitigation than LoYS and FC-CCDB, with similar performance to DDB. Notably, AracNet requires only 50 minutes on average for UrbanCars debiasing, including the backbone pre-training, monitors warmup and actual debiasing (on an Nvidia A100 GPU, with 20 GB of VRAM), while DDB requires the training of a diffusion model followed by the generation of synthetic bias-aligned samples, with several hours of computation.

4.4 AracNet provides insights on bias-propagation through the model

AracNet exploits shallow monitors to provide debiasing signals. Given a fixed architecture, rather than assuming the existence of a specific layer that universally encodes bias, we employ an automatic procedure for identifying the most biased layer according to the specific type of bias. Besides the indirect debiasing signals, as a byproduct, the monitors can provide insights on bias propagation throughout the model, which correlates with the type of bias. As an example, we report in Figure 2 confidence histograms of our monitors when debiasing Waterbirds and BFFHQ for 50 epochs. We can notice how in the first case (top), characterized by background bias, the monitor attached to the fourth layer shows, very soon, high confidence on all the training samples, while the inner layers, starting from the one attached to the first block, show an almost clear separation in confidence

Method	**ID.ACC** ↑	**BG-Gap** ↓	**CoObj-Gap** ↓	**BG-CoObj–Gap** ↓
ERM [22]	97.30	15.30	11.20	69.20
LLE* [22]	96.70	2.10	2.70	5.90
EIIL [7]	95.50	4.20	24.70	44.90
JTT [25]	95.90	8.10	13.30	40.10
LfF [30]	97.20	11.60	18.40	63.20
DebiAN [23]	**98.00**	14.90	10.50	69.00
LoYS [32]	<u>97.46</u>	8.08	9.86	39.98
DDB [6]	86.39	**1.85**	<u>0.52</u>	**0.12**
FG-CCDB [44]	92.98	4.17	7.37	4.90
AracNet (ours)	85.95 ± 1.01	<u>2.35 ± 0.63</u>	$\mathbf{-1.51 \pm 0.76}$	<u>0.35 ± 1.40</u>

Table 5: Results comparison on UrbanCars. We highlight in bold the best results from unsupervised debiasing works; the second-best are underlined.

between bias-aligned and bias-conflicting samples. Interestingly, this behaviour is different for BFFHQ (bottom), where the bias attribute is finer (gender). Here, the confidence of the first monitors reflects unpredictable features, the last layer, once again, shows training set memorization, while the third block seems to provide the best separation between conflicting and aligned subpopulations, showing that the best features to separate bias-aligned and bias-conflicting samples are of higher-level nature and reside in later stages of the model. Considering that our method is unsupervised, the per-monitor confidence distribution can provide additional information on the type of bias, allowing for a finer understanding of the bias captured by the model.

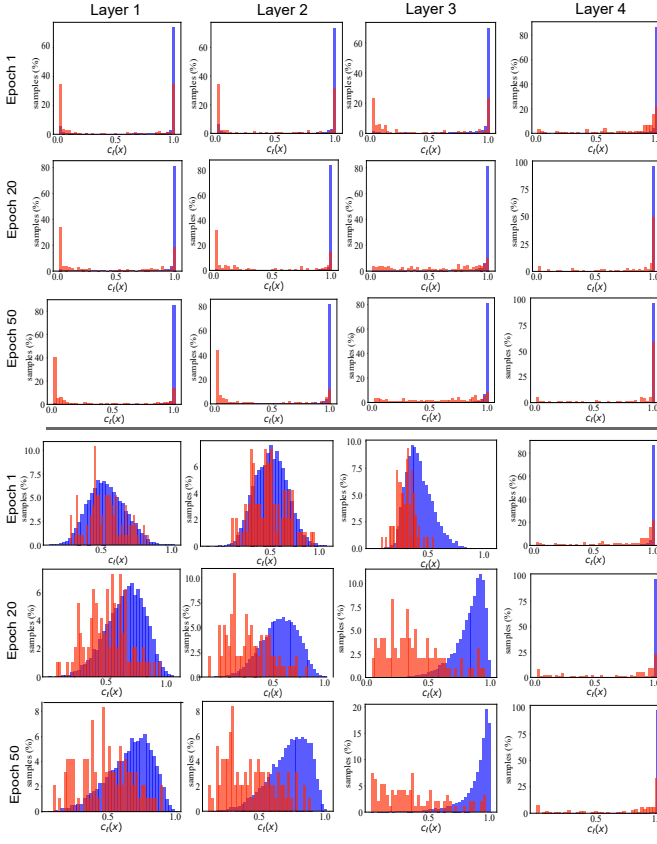

Fig. 2: Shallow Monitors' confidence histograms across 4 ResNet blocks after 1, 20, and 50 training epochs on Waterbirds (top) and BFFHQ (bottom). The last layer (closer to the network classifier) shows clear memorization patterns, with a single predominant peak where confidence is close to 1, regardless of the training epoch. On the other hand, it is possible to see how layers 1 and 2 show consistent separation between aligned and conflicting samples for Waterbirds, which is not the case for BFFHQ. The finer nature of gender bias seems to be better encoded in a later network stage (Layer 3).

Method	WGA
(AracNet w/o $w_{\text{surprisal}}$)	84.94 ± 1.41
AracNet w/o w_{ℓ}^{*}	88.78 ± 1.36
(Ours)	$\mathbf{91.27 \pm 0.56}$

(a) Loss weighting factors w_{l}^{*} and $w_{\text{surprisal}}$.

λ	WGA
1	89.63 ± 0.49
3	90.46 ± 0.34
4	88.05 ± 2.09
5	84.47 ± 2.58
2 (Ours)	$\mathbf{91.27 \pm 0.56}$

(b) Effect of λ.

#Epochs	WGA
50	$\mathbf{91.27 \pm 0.56}$
100	90.98 ± 0.21
200	90.85 ± 0.74

(c) Pretraining epochs for $f_{\theta^{\text{pre}}}$.

Table 6: Ablation studies for AracNet. All the ablations are measured on Waterbirds.

4.5 Additional results and ablation studies

In this Section, we perform additional experiments to evaluate the generality of AracNet while assessing the impact of its main components on the target model debiasing. All the ablation studies employ Waterbirds as the dataset.

Debiasing a vision transformer We design an experiment to assess whether our debiasing framework, based on monitors to probe bias across the model, can effectively generalize to vision transformers, where CNN's inductive bias is absent, with a different feature extraction mechanism. Specifically, we replicate our analysis on a ViT small with patch 16 (ViT-s/16), attaching a monitor at each transformer block (12, in this case). The results are reported in Table 2 and confirm how our monitors can provide effective debiasing also for transformer architectures, with a $\sim 23\%$ improvement in WGA over the ERM model.

The impact of $w_{\text{surprisal}}$ and w_{ℓ^*}. Here, we replicate our analysis removing either $w_{\text{surprisal}}$ or w_{ℓ^*} from AracNet's loss function. Table 6a summarizes the obtained results. A drop of $\sim 3\%$ corresponds to the usage of $w_{\text{surprisal}}$ only, while a more significant decrease in performance is observed when only using w_{ℓ^*}, confirming the benefit of employing both weighing factors in our loss.

The impact of λ. Table 6b shows the results obtained for different values of lambda when computing $w_{\text{surprisal}}$. When λ is set to 5, we observe a decrease of $\sim 7\%$ in the performance, while a more modest gap corresponds to the other investigated values, with best results for $\lambda = 2$.

Stability of AracNet debiasing with pre-training epochs. Here, we replicate our analysis by pre-training the backbone for a more extreme number of epochs, up to 200. As shown in Table 6c, even with 200 epochs, our monitors provide useful bias separation signals, leading to state-of-the-art performance on this dataset, confirming how the *where to look* protocol effectively solves the *when to stop* problem for unsupervised debiasing.

Transferring the Debiasing Signal to an External Model. While we generally intend our debiasing mechanism to work in a self-debiasing fashion, with the debiasing signal used to mitigate bias dependency directly on the same reinitialized backbone, this is not a strict rule per se. The per-sample reweighing scheme obtained from the selected shallow monitor can also be effective across different models, and no layer-to-layer correspondence is needed. In other words, the frozen backbone can be different from the model to debias. To verify the efficacy of this feature, we perform three runs using the selected best monitor from a ResNet18 biased backbone to debias a ResNet50 and a ViT-s-16, obtaining a WGA of 91.06 ± 0.36 and 90.68 ± 0.37, respectively.

5 Conclusion

This paper introduces AracNet, an unsupervised debiasing framework providing a solution to training conflicting memorization, exploiting effective debiasing signals extracted from specific intermediate layers for mining bias in vanilla models. By attaching shallow monitors to intermediate layers of a pre-trained network, we automatically identify the most biased layer and leverage its outputs to reweight samples, mitigating bias without requiring heuristic early stopping or annotated validation sets for avoiding conflicting memorization, with effective signals even after very long pre-training. Our approach exploits the observation that bias memorization varies across layers—early for background biases, later for semantic ones—enabling adaptive debiasing with negligible computational overhead. Experiments on standard benchmarks demonstrate that AracNet matches or surpasses state-of-the-art methods, offering a simple yet effective solution for real-world scenarios where bias labels are unavailable. Beyond its performance, AracNet's reliance on internal feature representations eliminates the need for external validation sets, addressing key limitations of prior work. This work not only advances unsupervised debiasing but also highlights the untapped potential of intermediate layers for understanding and correcting model biases. Future directions include extending our method to multi-modal biases and exploring dynamic monitor architectures for even greater adaptability.

Acknowledgements

This work is partially supported by Hi! PARIS and ANR/France 2030 program (ANR-23-IACL-0005).

References

1. Ahn, S., Kim, S., Yun, S.Y.: Mitigating dataset bias by using per-sample gradient. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=7mgUec-7GMv 2
2. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019) 3

3. Asaad, I., Shadaydeh, M., Denzler, J.: Gradient extrapolation for debiased representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3819–3829 (2025) 11
4. Capitani, G., Bolelli, F., Porrello, A., Calderara, S., Ficarra, E.: Clusterfix: A cluster-based debiasing approach without protected-group supervision. In: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 4858–4867 (2024). https://doi.org/10.1109/WACV57701.2024.00480 2
5. Ciranni, M., Molinaro, L., Barbano, C.A., Fiandrotti, A., Murino, V., Pastore, V.P., Tartaglione, E.: Say my name: a model's bias discovery framework. Transactions on Machine Learning Research (2025), https://openreview.net/forum?id=EuUxiiDZ7d 2, 4, 11
6. Ciranni, M., Pastore, V.P., Via, R.D., Tartaglione, E., Odone, F., Murino, V.: Diffusing debias: Synthetic bias amplification for model debiasing. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025), https://openreview.net/forum?id=66Z5tS8E45 2, 4, 9, 11, 12
7. Creager, E., Jacobsen, J.H., Zemel, R.: Environment inference for invariant learning. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 2189–2200. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/creager21a.html 12
8. Espinosa Zarlenga, M., Sankaranarayanan, S., Andrews, J.T., Shams, Z., Jamnik, M., Xiang, A.: Efficient bias mitigation without privileged information. In: European Conference on Computer Vision. pp. 148–166. Springer (2024) 2
9. Fang, T., Lu, N., Niu, G., Sugiyama, M.: Rethinking importance weighting for deep learning under distribution shift (2020), https://arxiv.org/abs/2006.04662 2, 4
10. Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020) 2, 3
11. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. Proceedings of the International Conference on Learning Representations (2019) 8
12. Hong, Y., Yang, E.: Unbiased classification through bias-contrastive and bias-balanced learning. Advances in Neural Information Processing Systems **34**, 26449–26461 (2021) 4
13. Jung, Y., Shim, H., Yang, J.Y., Yang, E.: Fighting fire with fire: Contrastive debiasing without bias-free data via generative bias-transformation. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 15435–15450. PMLR (23–29 Jul 2023), https://proceedings.mlr.press/v202/jung23b.html 4, 11
14. Kim, B., Kim, H., Kim, K., Kim, S., Kim, J.: Learning not to learn: Training deep neural networks with biased data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9012–9020 (2019) 2
15. Kim, E., Lee, J., Choo, J.: Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14992–15001 (2021) 4, 8, 9, 10
16. Kim, Y., Mo, S., Kim, M., Lee, K., Lee, J., Shin, J.: Discovering and mitigating visual biases through keyword explanation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11082–11092 (2024) 4, 9, 11

17. Kim, Y., Mo, S., Kim, M., Lee, K., Lee, J., Shin, J.: Discovering and mitigating visual biases through keyword explanation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11082–11092 (2024) 11
18. Krueger, D., Caballero, E., Jacobsen, J.H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., Courville, A.: Out-of-distribution generalization via risk extrapolation (rex). In: International conference on machine learning. pp. 5815–5826. PMLR (2021) 2
19. Lee, J., Kim, E., Lee, J., Lee, J., Choo, J.: Learning debiased representation via disentangled feature augmentation. In: Advances in Neural Information Processing Systems. vol. 34, pp. 25123–25133 (2021), https://proceedings.neurips.cc/paper/2021/file/d360a502598a4b64b936683b44a5523a-Paper.pdf 8, 10, 11
20. Lee, J., Park, J., Kim, D., Lee, J., Choi, E., Choo, J.: Revisiting the importance of amplifying bias for debiasing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 14974–14981 (2023) 2, 4, 9
21. Levy, D., Carmon, Y., Duchi, J.C., Sidford, A.: Large-scale methods for distributionally robust optimization. Advances in Neural Information Processing Systems **33**, 8847–8860 (2020) 2, 4
22. Li, Z., Evtimov, I., Gordo, A., Hazirbas, C., Hassner, T., Ferrer, C.C., Xu, C., Ibrahim, M.: 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. pp. 20071–20082 (2023) 4, 9, 12
23. Li, Z., Hoogs, A., Xu, C.: Discover and mitigate unknown biases with debiasing alternate networks. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 270–288. Springer Nature Switzerland, Cham (2022) 9, 11, 12
24. Lim, J., Kim, Y., Kim, B., Ahn, C., Shin, J., Yang, E., Han, S.: Biasadv: Bias-adversarial augmentation for model debiasing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3832–3841 (2023) 4
25. Liu, E.Z., Haghgoo, B., Chen, A.S., Raghunathan, A., Koh, P.W., Sagawa, S., Liang, P., Finn, C.: Just train twice: Improving group robustness without training group information. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 6781–6792. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/liu21f.html 2, 4, 7, 11, 12
26. Liu, S., Zhang, X., Sekhar, N., Wu, Y., Singhal, P., Fernandez-Granda, C.: Avoiding spurious correlations via logit correction. arXiv preprint arXiv:2212.01433 (2022) 11
27. Mehta, R., Roulet, V., Pillutla, K., Harchaoui, Z.: Distributionally robust optimization with bias and variance reduction. In: The Twelfth International Conference on Learning Representations (2024) 4
28. Nahon, R., De Moura Matos, I., Nguyen, V.T., Tartaglione, E.: Debiasing surgeon: fantastic weights and how to find them. In: European Conference on Computer Vision. pp. 435–452. Springer (2024) 2, 5
29. Nahon, R., Nguyen, V.T., Tartaglione, E.: Mining bias-target alignment from voronoi cells. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4946–4955 (October 2023) 4
30. Nam, J., Cha, H., Ahn, S., Lee, J., Shin, J.: Learning from failure: Training debiased classifier from biased classifier. In: Advances in Neural Information Processing Systems (2020) 1, 2, 4, 6, 7, 8, 9, 10, 11, 12

31. Pastore, V.P., Ciranni, M., Marinelli, D., Odone, F., Murino, V.: Looking at model debiasing through the lens of anomaly detection. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV). pp. 2548–2557 (February 2025) 2, 7, 9, 10, 11
32. Pastore, V.P., Ciranni, M., Murino, V.: Lose your self (loys): An adversarial entropy-based unsupervised approach for model debiasing. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5714–5723 (2026) 4, 9, 11, 12
33. Sagawa*, S., Koh*, P.W., Hashimoto, T.B., Liang, P.: Distributionally robust neural networks. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=ryxGuJrFvS 2, 4, 9
34. Seo, S., Lee, J.Y., Han, B.: Unsupervised learning of debiased representations with pseudo-attributes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16742–16751 (2022) 2
35. Shrestha, R., Kafle, K., Kanan, C.: Occamnets: Mitigating dataset bias by favoring simpler hypotheses. In: European Conference on Computer Vision. pp. 702–721. Springer (2022) 4
36. Sohoni, N., Dunnmon, J., Angus, G., Gu, A., Ré, C.: No subclass left behind: Fine-grained robustness in coarse-grained classification problems. Advances in Neural Information Processing Systems **33**, 19339–19352 (2020) 2, 4, 11
37. Tiwari, R., Shenoy, P.: Overcoming simplicity bias in deep networks using a feature sieve. In: International Conference on Machine Learning. pp. 34330–34343. PMLR (2023) 4
38. Vadakkeeveetil Sreelatha, S., Kappiyath, A., Chaudhuri, A., Dutta, A.: Denetdm: Debiasing by network depth modulation. Advances in Neural Information Processing Systems **37**, 99488–99518 (2024) 9, 10, 11
39. Zhang, M., Sohoni, N.S., Zhang, H.R., Finn, C., Ré, C.: Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. arXiv preprint arXiv:2203.01517 (2022) 4
40. Zhang, M., Sohoni, N.S., Zhang, H.R., Finn, C., Re, C.: Correct-n-contrast: a contrastive approach for improving robustness to spurious correlations. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 26484–26516. PMLR (17–23 Jul 2022), https://proceedings.mlr.press/v162/zhang22z.html 11
41. Zhang, Z., Feng, M., Li, Z., Xu, C.: Discover and mitigate multiple biased subgroups in image classifiers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10906–10915 (June 2024) 4
42. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems **31** (2018) 4
43. Zhao, M., Li, C., Zhang, Q.: Class-conditional distribution balancing for group robust classification. arXiv preprint arXiv:2504.17314 (2025) 9, 11
44. Zhao, M., Zhang, Q.: Fine-grained class-conditional distribution balancing for debiased learning. In: The Fourteenth International Conference on Learning Representations 11, 12