Review of the state of the art of algorithmic debiasing schemes for Deep Learning

Nathan Roos
Télécom Paris
nathan.roos@telecom-paris.fr

January 23, 2025

Abstract—**This work is the first production for the "Introductory Research Track" course on the subject "Algorithmic debiasing schemes for Deep Learning", followed by Nathan Roos under the supervision of Enzo Tartaglione.**

Contents

I Introduction 1

II The problem of bias 1
II-A Formal definitions 1
II-A1 Definitions and notations 1
II-A2 Debiasing is not Fairness 2
II-A3 Cross-bias generalization vs Isolated-bias generalization 3
II-A4 Common DNN vocabulary 3
II-B Why is bias a problem ? 3
II-B1 Fairness issues : an ethical problem 3
II-B2 Bad generalization causes unreliability . . 3
II-B3 Legal Imperatives 3

III Datasets for debiasing 4
III-A Benchmarks based on MNIST 4
III-B Benchmarks based on CIFAR-10 5
III-C CelebA 5

IV Debiasing techniques 5
IV-A Bias extraction 6
IV-A1 Training the Bias Amplifier 6
IV-A2 Extracting the bias information from the BA 7
IV-B Bias mitigation 7
IV-B1 Pre-processing methods 7
IV-B2 Post-processing methods 8
IV-B3 In-processing methods 8
IV-C Comparison 10

V Biasedness metrics 11
V-A Accuracy-based metrics 11
V-B Fairness metrics 12
V-C Estimating biasedness by characterizing the mistakes 12
V-D Bias discovery 13

VI Conclusion: future research directions 13

VIIAppendix 16
VII-ASKEWSIZE: computing Cramér's V 16
VII-BComprehensive comparison 16
VII-CMore benchmark datasets 16
VII-C1 Waterbirds 16
VII-C2 BAR 16
VII-C3 Benchmarks based on ImageNet 16

I. Introduction

Deep Neural Networks (DNN) are nowadays famous for their performance on many computer vision tasks, such as classification, semantic segmentation and object detection. They are being deployed in high stakes applications, such as autonomous driving, face recognition, medical imaging, augmented reality, robotics, etc.

Formerly, the focus of researchers was on making these models accurate. Now that they achieve state of the art performance, a new challenge arises. Indeed, DNN are prone to rely on *spurious correlations* or *biases*, *i.e.*, correlations that hold for the training data but not in general, a phenomenon that has been sometime called shortcut learning [29].

As a first example, consider the image classification task of detecting pedestrians. If environment cues such as the presence of a sidewalk is correlated with the target class "pedestrian" in the training data, DNN may exploit the environment cues for classification, leading to performance degradation on images containing a pedestrian but no sidewalk [29].

In the next section, we will provide a formal definition of the bias, as well as an in-depth explanation of why biases are a problem. In the third section, we will present three commonly used datasets, which will illustrate the type of situation that debiasing can tackle. In the fourth section, we will study the existing debiasing techniques. In the fifth section, candidate biasedness metrics will be presented. As a conclusion, we will see the possible research directions.

II. The problem of bias

A. Formal definitions

1) Definitions and notations: For the rest of this review, we define the following variables :

- X a random variable of domain $\mathcal{X}$ representing the input distribution of the data samples,

- Y a discrete random variable of finite domain $\mathcal{Y}$ and cardinality $N_Y \triangleq |\mathcal{Y}|$ which represents the *target* attribute (or *class* attribute) associated with each sample[1],
- B a discrete random variable of finite domain $\mathcal{B}$ and cardinality $N_B \triangleq |\mathcal{B}|$ which represents the *spurious* attribute (or *bias* attribute),
- $\mathcal{D} \triangleq \{(x_i, y_i, b_i)\}_{i=1}^{N}$ the dataset of N samples used to train the classifier (such that $\forall i \in [|1; N|], (x_i, y_i, b_i) \overset{\text{iid}}{\sim} (X, Y, B)$),
- $\mathcal{M} : \mathcal{X} \longrightarrow \mathcal{Y}$ a classifier trained to predict Y from X,
- $R \triangleq \mathcal{M}(X)$ the output of the classifier $\mathcal{M}$, of domain $\mathcal{R}$ (depending on the context, it can be a class label $y \in \mathcal{Y}$ or a probability distribution on $\mathcal{Y}$),
- (X^*, Y^*, B^*) the underlying ground truth distribution we aim to learn by debiasing.

In Debiasing[2], we consider the situation where the target variable Y is spuriously correlated with the bias attribute B. That is, the relationship between Y and B is different between the data distribution (X, Y, B) and the underlying true distribution (X^*, Y^*, B^*). Crucially, we do not have access to the underlying true distribution. To formalize the relationship between Y and B, we introduce the Normalized Mutual Information as a tool to assess the similarity of the partitioning of a dataset obtained using different attributes.

Definition 1 (Normalized Mutual Information (NMI))**.** Let A and B be two discrete random variables of finite entropy such that A or B is non-deterministic. The Normalized Mutual Information (NMI) between A and B is defined as follows:

$$\text{NMI}(A; B) \triangleq \frac{2 \cdot I(A; B)}{H(A) + H(B)}$$

Remark 1. Using the properties of the mutual information, it follows that $\text{NMI}(A; B) \in [0; 1]$ and $\text{NMI}(A; B) = 0 \iff A \perp B$.

Therefore, in Debiasing, we consider the setting where the following working hypothesis holds:

$$\text{NMI}(Y; B) \neq \text{NMI}(Y^*; B^*) \tag{WH1}$$

In fact, in most of the debiasing literature, the working hypothesis is even stronger :

$$\text{NMI}(Y; B) > \text{NMI}(Y^*; B^*) = 0 \tag{WH2}$$

In other words, we assume that there is a spurious correlation between Y and B in the dataset which is not representative of the underlying true distribution where $Y^* \perp B^*$ (using remark 1). We will consider ourselves under (WH2) until section V where we will assess the proposed metrics to measure the bias.

Definition 2 (Biased classifier)**.** We say that a classifier $\mathcal{M}$ trained on a dataset $\mathcal{D}$ drawn from the distribution (X, Y, B) and producing an outcome R is *biased* towards the attribute B compared to the underlying distribution (X^*, Y^*, B^*) iff:

$$\text{NMI}(R; B^*) \neq \text{NMI}(Y^*; B^*)$$

Using the hypothesis made just above, it is equivalent to:

$$R \not\perp B^*$$

Intuitively, it means that a biased classifier relies on the bias attribute to make its prediction.

The goal of Debiasing is to find methods to train classifiers that are not biased.

Definition 3 (Bias label related to a class)**.** Let $(y, b) \in \mathcal{B} \times \mathcal{Y}$ and $\epsilon > 0$. We say that b *is a bias label* ϵ*-related to class* y if:

$$P(Y = y | B = b) > P(Y = y) + \epsilon$$

The previous definition uses the ϵ-margin to highlight that the difference between $P(Y = y | B = b)$ and $P(Y = y)$ should be significant (for instance $\epsilon > 0.05$). From now on, we will omit the ϵ.

Definition 4 (Bias-aligned and bias-conflicting samples [30])**.** Let $(x, y, b) \in \mathcal{D}$ be a sample, its class label and its bias label. We say that the sample x is

- *bias-aligned*[3] if b is related to class y
- *bias-conflicting*[4] if b is not related to class y

Examples of bias-aligned and bias-conflicting samples are provided in section III.

2) Debiasing is not Fairness: Debiasing is an area of research close to Fairness, but different, and the two should not be confused.

Fairness in machine learning aims at balancing the output of a model with respect to a sensitive bias attribute B [38]. X is typically an individual's attributes, and the bias attribute B the gender, the age, the ethnicity, etc. This approach focuses on enforcing fairness metrics so that the outcomes of a classifier ensure *equal treatment* or *outcomes* across the different demographics defined by B (fairness metrics will be presented in section V-B). Crucially, the explicit knowledge of how X maps to Y given B is not required, or even relevant.

On the other hand, Debiasing seek to correct the influence of B on the relationship between X and Y, so that it is representative of the influence of B^* on (X^*, Y^*) (with the common hypothesis $Y^* \perp B^*$).

Hence, the goals of Fairness and Debiasing are different. The goal of debiasing is to make sure that the model's predictions are not unduly influenced by B, *i.e.*, we want the classifier to learn the underlying ground truth distribution. Conversely, the goal of Fairness is to make sure that the model's outcomes ensure equal treatment for all groups, whatever the ground truth distribution might be.

[1] $\triangleq$ denote the definition of a quantity.

[2] We will use "Debiasing" with a capital D to refer to the topic of research.

[3] in some works, bias-aligned becomes "bias-guiding" [22, 18]

[4] in some works, bias-conflicting becomes "bias-misaligned" [29, 9, 28]

3) Cross-bias generalization vs Isolated-bias generalization: Unlike previous works, we identity two distinct types of debiasing scenarios which are found in the literature.

The first one is the problem of *cross-bias generalization*, as coined by Bahng *et al.* [6]. In this scenario, the bias-conflicting samples of a given class have bias information related to another class (*i.e.*, their bias labels are related to another target class), as illustrated in figure 1. Benchmark datasets corresponding to that definition are Biased-MNIST, CelebA, Corrupted CIFAR-10 (cf III-A, III-C, III-B).

In the second scenario, bias-conflicting samples from a given class do not have bias information related to another class. We coin this setting *isolated-bias generalization* and illustrate it in figure 2. The dataset corresponding to that definition is BAR (cf VII-C2 for an example).

4) Common DNN vocabulary: In this section, we recall the common vocabulary used by the deep learning community that we will use in the next sections[5].

A Deep Neural Network (DNN) is organized as several layers of neurons sequentially connected. In the context of image classification, the DNN $\mathcal{M} : \mathcal{X} \longrightarrow \mathbb{R}^{N_Y}$ takes as input an image from the domain $\mathcal{X}$ and returns a score for each target class (which can be transformed into a probability distribution over the class labels by applying the softmax function). The model can be divided into two parts, both from the architectural and functional point of view.

The first part of the DNN is referred to as *encoder* or *backbone*. It maps a given image into a space with much fewer dimensions than $\mathcal{X}$. From the architectural point of view, the encoder often consists of all layers but the last few linear layers. The last layer of the encoder is called the *bottleneck* layer, containing K neurons. From the functional point of view the encoder can be written as $\mathcal{E} : \mathcal{X} \longrightarrow \mathbb{R}^K$. The output of the encoder $\mathcal{E}(x)$ is often referred to as the *bottleneck representation* or *latent representation* or *latent attributes* of $x \in \mathcal{X}$ while $\mathbb{R}^K$ is called the *latent space* or *feature space* [28]. The second and last part of the DNN is referred to as *decoder* or *classification head* or *classifier*. It often consists of the very last few linear layer of the network (*e.g.* VGG [25] and ResNet [13] architecture). From a functional standpoint, it writes as $\mathcal{C} : \mathbb{R}^K \longrightarrow \mathbb{R}^{N_Y}$. The role of the classification head consists in partitioning the feature space into each of the different classes. Overall, the model can be written as $\mathcal{M} = \mathcal{C} \circ \mathcal{E}$ and is summarized in figure 3.

B. Why is bias a problem ?

Bias is a problem from both an ethical standpoint, and from a reliability perspective. It is now also the focus of the law.

1) Fairness issues : an ethical problem: We underlined before that the field of Debiasing and Fairness are different. However, it is still possible to apply debiasing methods to fairness issues, if the goal is to make sure that the outcome of a classifier is not unduly influenced by a sensitive attribute (rather than forcing the outcomes of the classifier to be balanced with respect to a sensitive attribute).

Biases occur in facial recognition models trained predominantly on images of light-skinned individuals, as those models performed poorly on darker-skinned individuals. Similarly, large language models trained on a large corpus of text (including a large part of text from the Internet) often learn and reproduce gender and racial stereotypes present in the training data.

Biases cause fairness issues, and therefore constitute an ethical problem.

2) Bad generalization causes unreliability: The presence of biases in datasets is not *always* a problem. They only cause performance issues if the classifier uses them to complete its task.

Following [30], we say that B is *malignant* (resp. *benign*) when a model trained on $\mathcal{D}$ suffers (resp. does not suffer) performance degradation on an unbiased evaluation set compared to a model trained on an unbiased dataset. Nam *et al.* [30] argue that the bias being *malignant* imply that it is easier for the model to learn the bias, and therefore the bias attribute is used as a proxy by the model to predict the target attribute (hence the term *shortcut learning* [28]). However, when the bias is *benign* it means that the bias attribute is as hard or harder to learn than the target attribute, and therefore the classifier does not use the bias to complete its task.

It is useful to keep in mind this distinction, because biases are extremely frequent, but they constitute a problem and should be treated with debiasing methods only if they are malignant. In section III-C, we provide a real world example based on the CelebA [26] dataset.

In the presence malignant bias, the model will rely on the bias attribute to predict the target class. This leads to a significant drop in accuracy on samples where the correlation between the bias and the target attributes does not hold. Hence, the model will badly generalize to unbiased datasets (which represents the true distribution on which the model will be applied if deployed in the real world), and therefore be unreliable.

3) Legal Imperatives: To address the growing concerns regarding AI's ethical and reliability issues, several regulations have been enforced by the EU. The General Data Protection Regulation (GDPR) is effective since May 2018. It grants to human users the right to an explanation for algorithmic decision-making, which applies to AI [33]. The EU Artificial Intelligence Act (AI Act) is a piece of legislation that came into force in 2024 aiming to classify and regulate artificial intelligence applications based on their risk of causing harm [34]. It emphasizes the need for warranties to ensure safety and fairness.

In the European Union, it is therefore a legal imperative to be able to explain a decision made by an AI about a user, which encompasses the ability to characterize how much a given sensitive bias attribute came into play. It is also mandatory to put in place measures (such as debiasing methods) to ensure that AI systems do not cause harm.

Although both the GDPR and the AI Act are European laws, they have the potential to establish a global benchmark,

[5] The vocabulary presented here is the same for the rest of the report. However, all the notations introduced in this section are only there for clarity and will not necessarily be used as such afterwards.

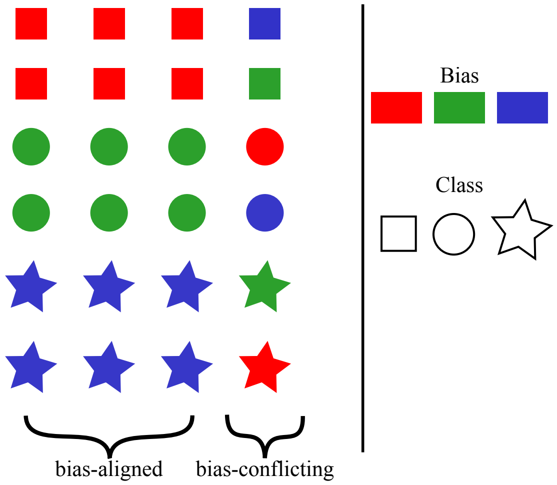

Fig. 1. Cross-bias generalization scenario

because of the phenomenon known as the Brussels effect [8].

III. Datasets for debiasing

In this subsection, we will briefly describe the most commonly used[6] benchmark datasets used to assess the performance of debiasing methods in the context of image classification.

We will focus on computer vision datasets, but it should be noted that debiasing also applies to the field of Natural Language Processing (NLP). For instance, in Natural Language Inference (NLI) where the task is to determine whether a given "hypothesis" sentence is entailed by, neutral with, or contradicts a given "premise" sentence, some works [11] have shown that datasets such as MultiNLI present spurious correlation between contradictions and the presence of negation words.

Most of the image datasets are obtained by using an already existing collection (*e.g.* , MNIST, CIFAR-10) and applying transformations to them to make them biased with respect to controlled attributes. We will call "natural image" a picture of the real world that has been obtained with a camera[7]. We will call "synthetic" a dataset or an image obtained after applying transformations to an image.

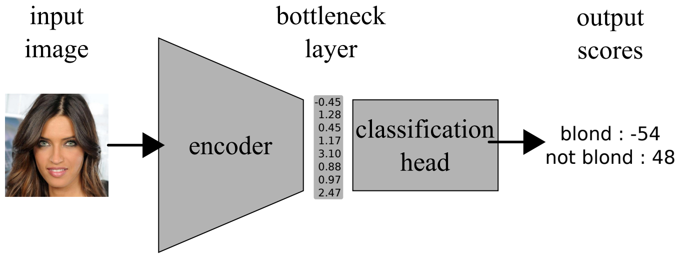

Fig. 3. Standard architecture organization for classification tasks in computer vision.

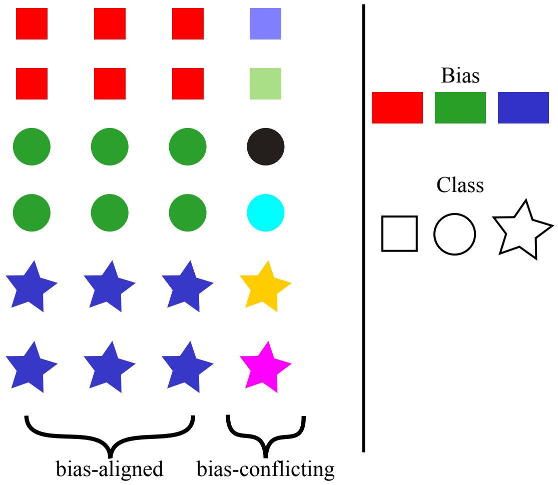

Fig. 2. Isolated-bias generalization scenario

For the sake of conciseness, the descriptions of Waterbirds, BAR and ImageNet are in the Appendix VII-C

A. Benchmarks based on MNIST

The MNIST dataset [20] is a collection of grayscale images of handwritten digits from 0 to 9 ($\mathcal{Y} = [|0;9|]$), with a resolution of 28x28. It contains 60,000 images for training and 10,000 for testing. Due to its simplicity, MNIST is used as a base for several synthetic debiasing benchmarks. For the sake of brevity, we will only focus on the Biased MNIST dataset (see Fig 4 for examples) used for the first time in [6], but the process is similar for the other MNIST-based synthetic benchmarks (Colored MNIST [21], Multi-color MNIST, Multi-bias MNIST). The process is the following: first we pick a color for each of the 10 classes (*e.g.* , 0: red, 1: green, etc). Then, for a proportion ρ of the training images, the color associated to the class is injected in the background (in Biased MNIST [6], the color is injected in the digit), and for the remaining proportion $1 - \rho$ of the training images, the color of another class is injected in the background. In the test set, two settings are possible : either the color is injected in the background independently of the class of the image (this yields the *unbiased* test set) or the color injected in the background is different from the color associated to the class of the images (this yields the *bias-conflicting* test set). The parameter ρ is used to control the correlation we want between the target attribute (digit) and the bias attribute (color) in the training data. ρ and is typically taken in the range $\{0.99, 0.995, 0.999\}$. It allows to synthetically simulate increasing levels of difficulty. In the case of Biased MNIST, a single source of bias is added, but other benchmarks simulate multiple sources of bias. For instance, the Multi-color MNIST dataset injects 2 colors in the background of each digit, and the Multi-bias MNIST dataset add 7 bias attributes : digit color, fashion object, fashion color, Japanese character, Japanese character color, English character, and English character color.

[6] Specifically, the benchmarks presented are those used to evaluate the methods that will be presented below.

[7] In particular, a "natural image" is *not necessarily* a picture of nature.

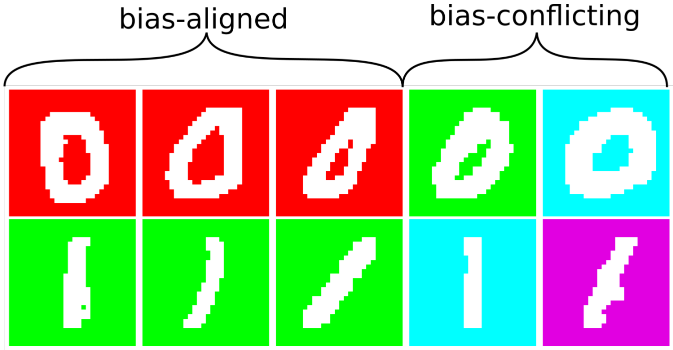

Fig. 4. Examples from the Biased MNIST dataset. The first (resp. second) row are samples from the target class 0 (resp. 1). The first 3 columns are bias-aligned samples (*i.e.*, samples having their bias attribute correlated with their target attribute) and the last 2 columns are bias-conflicting samples.

B. Benchmarks based on CIFAR-10

CIFAR-10 is a dataset of natural images introduced in [19]. It contains 60,000 32x32 color images from 10 different classes: $\mathcal{Y} = \{$airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck$\}$. Despite the low resolution of the images, CIFAR-10 is a widespread choice for evaluating the properties of newly developed methodologies in computer vision and machine learning. The biased datasets build upon CIFAR-10 rely on previous works by Hendrycks *et al.* [14] where different types of corruption were added to the test set images with the goal to measure the resilience of classifiers as the severity of the corruption increases. The 20 possible type of corruptions are $\mathcal{B} = \{$Snow, Frost, Fog, Brightness, Contrast, Spatter, Elastic, JPEG, Pixelate, Saturate, GaussianNoise, ShotNoise, ImpulseNoise, SpeckleNoise, GaussianBlur, DefocusBlur, GlassBlur, MotionBlur, ZoomBlur, Original$\}$. They aim to emulate realistic corruptions that images could suffer from. To create the synthetic Corrupted CIFAR-10 dataset, one corruption is selected per class, then the corruptions are applied to all images in the dataset such that a proportion ρ of the training images present a correlation between the type of corruption and the target class (see Fig. 5 for examples). ρ plays the same role as for the MNIST based benchmarks (cf section III-A). At least three variants of the dataset are in use, depending on the corruption chosen and the level of severity. They are denoted Corrupted CIFAR-10^{0,1,2}.

C. CelebA

Introduced in [26], CelebA is a face-attributes recognition dataset containing more than 200,000 natural images with 40 face attribute annotations (*e.g.* , "Eyeglasses", "Blond Hair", "Perceived gender", "Mustache", etc). Any attribute can be the target of a classification task. However, depending on the selected target attribute, the subgroups that appear in the classes can be heavily unbalanced, and thus biases appear naturally. For instance, when choosing as target attribute "Blond hair", the "Perceived gender" attribute becomes a bias attribute because a majority of people without blond hair are males. This makes CelebA a useful benchmark to test the performance of debiasing techniques on "natural" biases (*i.e.*, not synthetic).

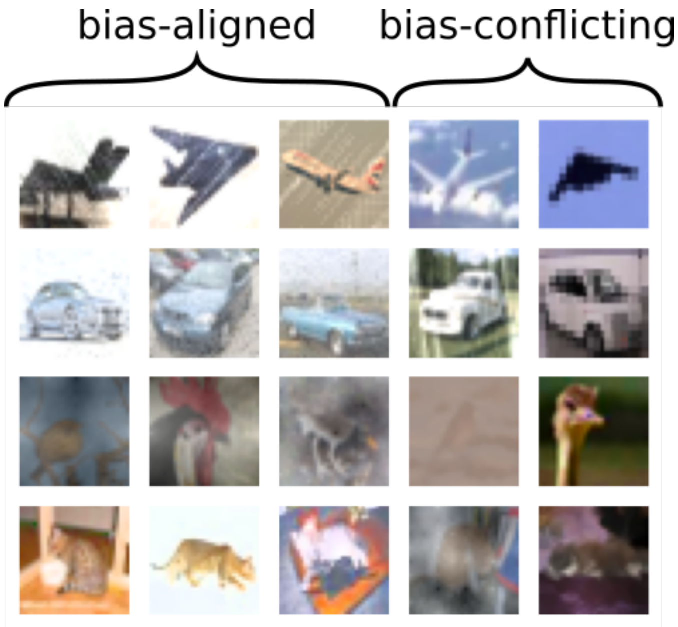

Fig. 5. Examples from the Corrupted CIFAR-10 dataset. From top to bottom, the rows are samples from the classes : airplane, car, bird, cat. The first three columns are bias-aligned images, *i.e.*, their corruption is the one correlated with their class, while the last two columns are bias-conflicting images : their corruption is correlated to another class.

IV. Debiasing techniques

In the Debiasing literature, the techniques are often [28, 22, 1] organized as follows:

1) **Supervised** methods : the labels for the bias feature B are supposed available in the training data. The explicit correction of B's influence on predictions is then possible.
   a) **Pre-processing** methods : the dataset is modified before classification.
   b) **In-processing** methods : the learning process is tweaked (change of the optimization algorithm and/or of the objective function and/or of the regularization constraints).
   c) **Post-processing** methods : the output of the DNN is modified.
2) **Unsupervised** methods : the bias labels are assumed absent from the training data. This corresponds to a realistic setting because bias labels are often difficult or expensive to obtain in real use cases.
   a) **Bias-tailored** methods (or **weakly supervised** methods, or **prior-guided** methods): there is prior information about the nature of the bias.
   b) **Unsupervised agnostic** methods : neither the bias labels nor prior information about the bias are available.

However, most unsupervised agnostic methods can actually be divided into two steps[8]: first a *bias extraction* step, where information about the bias is inferred from the data or from the training dynamics of a model trained on a biased dataset. The second step is *bias mitigation*, where the methods employed are similar to those traditionally put in the supervised in-processing category. We argue that treating separately supervised and unsupervised methods is detrimental because it obscures the available design space by framing unsupervised methods as inseparable systems, while it could be efficient to combine the bias extraction technique of an unsupervised method and the bias mitigation technique of another method (supervised or unsupervised). This is similar to the approach of the problem of invariant learning (learning domain invariant representations) as formulated by the authors of [10], where any environment inference technique can be combined with any invariant learning pipeline. This is why we will separate our study of debiasing methods into the analysis of bias extraction and of bias mitigation methods in the following manner:

1) **Bias extraction** methods.
    a) Training a biased model called "bias amplifier" (BA)
    b) Extracting bias information from the BA
2) **Bias mitigation** methods.
    a) **Pre-processing** methods
    b) **Post-processing** methods
    c) **In-processing** methods.

The goal here is not to make an exhaustive list of the existing methods, but rather to focus on a few representative methods and to analyze and describe them precisely.

We introduce a few notations in this section[9]:

- θ denotes the parameters of the debiased model,
- ϕ denotes the parameters of the BA,
- f_θ denotes the debiased model,
- g_ϕ denotes the BA model,
- $p(x;\theta)$ denotes the softmax output of the model with parameters θ when x is given as input,
- $p_y(x;\theta)$ denotes the probability assigned to the target class y by the model with parameters θ when x is given as input,
- $\hat{y}$ will denote the class predicted by the model given an input sample x.

A. Bias extraction

Bias extraction methods often rely on a biased model. We will call it the *bias amplifier* (BA). It is designed to *capture* or *attach to* the bias (*i.e.*, we want the BA to use the bias attribute to make its prediction). The BA can be simply a vanilla ERM (Empirical Risk Minimization) model trained with cross-entropy (CE) loss on a biased dataset. Many methods try to influence the learning process to make the BA more biased than a vanilla ERM.

1) Training the Bias Amplifier:

a) Bias is learned early: In [24] and [30], the authors rely on the assumption that bias features are easy to learn, and therefore are learned early, before the intrinsic features are learned (*i.e.*, the features of the object corresponding to the target class).

In Just Train Twice (JTT) [24], the authors assume they have a small validation set with bias labels and a large training set without bias labels. They build on the observation that sufficiently low complexity Empirical Risk Minimization (ERM) models tend to fit bias-aligned samples. Therefore, by training such a model for a fixed small number of training steps T, they get a biased model. T is treated as a hyperparameter which is tuned so that the debiased model obtained in the end achieve a maximal bias-conflicting validation accuracy.

In Learning from Failure (LfF) [30], the authors also make the assumption that bias features are easy to learn and therefore are the first features to be learned. Thus, the model is encouraged to learn a lot from the first samples it correctly classifies, assuming they are bias-aligned and that this will lead the model to learn to use the bias features. To this end, they use the generalized cross entropy (GCE) loss introduced in [42]:

$$\text{GCE}(p(x;\phi), y) = \frac{1 - p_y(x;\phi)^q}{q}, \quad \forall\ (x,y) \in \mathcal{D}$$

where $q > 0$ is a hyperparameter. Compared to the CE loss, the gradient of the GCE up-weights the gradient of the CE loss for the samples on which the model predicts the correct class with high probability:

$$\frac{\partial \text{GCE}(p(x;\phi), y)}{\partial \phi} = (p_y(x;\phi))^q \frac{\partial \text{CE}(p(x;\phi), y)}{\partial \phi}$$

Using the GCE causes the correct samples classified with high confidence by the model (*i.e.*, the probability of the target class is high) to have high gradient. Under the assumption that the first samples to be correctly classified are the bias-aligned samples, it causes them to weight more in the learning process and therefore cause the model to learn biased features.

b) Bias is not learned early: In VCBA[10] [28], the authors do not make the assumption that bias features are easy to learn and therefore learned early. Instead, they make the weaker hypothesis that at some point during training (not necessarily at the beginning), the model uses the bias cues to make its prediction. The intuition as to why this hypothesis is better is that it could be advantageous in cases where we know that multiple biases are at stake and that they are learned at different rates like in the Multi-color MNIST dataset. To detect the moment when the BA g_ϕ uses the most the bias, they use the latent representation of the samples in the bottleneck layer. For every training epoch e, the classifier $\mathcal{M}$ maps each input sample x_n to a vector of latent attributes $\mathbf{a}_{e,n} \in \mathbb{R}^K$ (where K is the number of neurons in the bottleneck layer). For each target

[8] This is true for [24, 28, 10]. For [21] and [30], it is admittedly not straightforward since the steps are performed simultaneously. However, after a slight transformation, both methods fit in our 2-steps framework.

[9] We will frequently need other notations for specific cases, which will be introduced when required.

[10] acronym: Voronoi Cells Bias Alignment

class $y \in \mathcal{Y}$ and each epoch e, we define the class centroid $\mathbf{C}_{e,y}$ as the average of the bottleneck representation of the correctly classified samples of class y. We further define the Voronoi boundary $\mathbf{H}_{e,i,j}$ as the hyperplane equidistant from $\mathbf{C}_{e,i}$ and $\mathbf{C}_{e,j}$. The authors assume that the farther a misclassified sample is from its target Voronoi cell, the more it has been strongly pulled by an attractor resulting from some bias. Hence, when the average distance between the misclassified samples and their target Voronoi cells reaches its maximum, the BA is deemed to have learned bias features. It happens at epoch e^* defined as follows:

$$e^* \triangleq \underset{e}{\operatorname{argmax}} \frac{\sum_n d^*(\mathbf{a}_{e,n}) \cdot \overline{\delta}_{\hat{y}_{e,n},y_n}}{\sum_n \overline{\delta}_{\hat{y}_{e,n},y_n}}$$

where $\overline{\delta} = 1 - \delta$, δ is the Kronecker delta, $\hat{y}_{e,n}$ is the class predicted by the BA at epoch e and:

$$d^*(\mathbf{a}_{e,n}) \triangleq d(\mathbf{a}_{e,n}, \mathbf{H}_{e,i,j}) \cdot \frac{2}{\left\|\mathbf{C}_{e,\mathcal{M}(x_n)}\right\|_2 + \left\|\mathbf{C}_{e,y_n}\right\|_2}$$

The first term of the product denotes the ℓ_2 distance between $\mathbf{a}_{e,n}$ and $\mathbf{H}_{e,i,j}$ and the second one is a scaling term that accounts for the fact that in the latent space, the distances tend to shrink because of the weight decay.

Although the authors do not refer to JTT [24] explicitly, this method can be seen as building up on JTT where instead of fine tuning the number of steps after which we stop the training using an annotated validation set, we use a metric to detect when the bias is maximally used.

Drawing inspiration from VCBA, Ciranni *et al.* [9] detect the moment when the model fits the most the bias by examining the space of the softmax outputs. For all sample x_n and training epoch e, they define a metric indicating how far the n-th sample is from being correctly classified:

$$d_{e,n} = \begin{cases} \max(p(x_n;\phi_e)) - p_{y_n}(x_n;\phi_e) & \text{if } \hat{y}_{e,n} \neq y_n \\ 0 & \text{else} \end{cases}$$

where $p(x_n;\phi_e)$ is the softmax output of the BA g_ϕ given x at epoch e. Then they find the epoch e^* when the model confidently misclassifies a pool of samples (supposedly biased):

$$e^* \triangleq \underset{e}{\operatorname{argmax}} \frac{1}{\sum_n \overline{\delta}_{\hat{y}_{e,n},y_n}} \sum_n d_{e,n}$$

where $\overline{\delta} = 1 - \delta$ and δ is the Kronecker delta. The advantages over VCBA of working at the output of the model instead of in the latent space are twofold. First, it saves computation, since the latent space has typically a higher dimensionality and because it is not required to compute the class centroids nor the hyperplanes. Second, it is more interpretable, because the metric measures how much the model is confident in misclassifying the (supposed) bias-conflicting samples.

In LWBC[11] [18], the BA is trained for a few epochs on a biased dataset, like JTT. Their approach is original in that they also update the BA during the training of the debiased model. Since the BA is used to infer which samples are bias-conflicting and deserve to be focused on, as the debiased model becomes better, it might be useful to also update the BA so that it only misclassifies the most difficult bias-conflicting samples. In order to update the BA, the authors propose to use knowledge distillation, by distilling the knowledge of the debiased model into the BA.

2) Extracting the bias information from the BA: Given a BA, trained using the previous techniques, there are multiple ways to extract bias information. One technique is to infer bias pseudo-labels. In VCBA [28], the samples misclassified by BA are labeled bias-conflicting, whereas the samples correctly classified are labeled bias-aligned.

In LWBC [18], the authors build upon the insight that BA models can be sensitive to hyperparameter choices. Therefore, they propose to bootstrap a biased committee by training m BAs on m different subsets of the dataset (sampled with replacement). Every subset is guaranteed to be dominated by bias-aligned samples, and therefore the BAs all learn the bias features, while still being different. Then, instead of a binary bias pseudo-labels, we can assign to every sample $x \in \mathcal{X}$ a level of bias as the proportion of BAs that classified x correctly.

Another approach is proposed in EIIL[12] [10], where the authors build upon the objective function of IRMv1 [5] whose aim is to learn representations invariant across environments (here, the environments are the groups of samples sharing the same bias attribute) to infer the environments (and thus the bias-conflicting samples) in a unsupervised way. Namely, the environments inferred are those that maximally violates the invariance principle IRMv1.

As we will see further in section V, one gap in the debiasing literature is that the biasedness of a model is rarely considered or evaluated at all in the case of bias extraction methods. This leads to an impossibility to directly compare the aforementioned methods. Therefore, bias extraction methods can only be compared through performance metrics when they are integrated into a pipeline {bias extraction->bias mitigation}.

B. Bias mitigation

1) Pre-processing methods: The intuitive approach when facing with the bias is to simply remove the bias feature from the dataset, it is called *fairness by blindness*. However, there are two problems with this method. First, it might not be possible to do in the case of images; for instance, removing the bias feature in the Corrupted CIFAR-10 dataset would mean restoring the images to their original state, starting from the corrupted version. Second, this approach is ineffective because of the phenomenon of *encoding redundancy* [12], which states that information is very rarely encoded only once in the data, and thus that a bias attribute can be deduced from other features [31].

Since the problem of the dataset being biased is the over-representation of bias-aligned samples vs bias-conflicting

[11] acronym: Learning With Biased Committee

[12] acronym: Environment Inference for Invariant Learning

samples, another simple solution can be to manually add new bias-conflicting samples. However, due to the often prohibitive cost of gathering new data (especially bias-conflicting, which might be rare occurrences), this approach is not followed in practice. Nonetheless, it is still possible to simply emulate the effect of adding bias-conflicting samples to the dataset by oversampling them. The problem here is that this approach is actually equivalent to up-weighting bias-conflicting samples in the loss function [24, 30, 28], except that oversampling is much more computationally expensive. Reweighting methods will therefore be preferred.

Data augmentation can also be used to improve the dataset. In particular, it is a solution to the problem that reweighting techniques tend to make the model overfit on the bias-conflicting samples when there are just a few of them. In BiasAdv [22], the authors generate synthetic bias-conflicting samples by attacking the prediction of the bias amplifier model (BA) g_ϕ. However, since the BA is not perfect, the generated image might not resemble a natural image. Therefore, the authors add the constraints that the attack must not change the prediction of the debiased model being trained f_θ. Formally, given a sample $(x, y) \in \mathcal{D}$, an adversarial image x_{adv} is generated by solving:

$$x_{adv} = \underset{\tilde{x} \triangleq x+\epsilon}{\arg\max} \left[\mathcal{L}(\tilde{x}, y; \phi) - \lambda \cdot \mathcal{L}(\tilde{x}, y; \theta)\right]$$

where $\mathcal{L}$ denotes the CE loss and $\lambda > 0$ is a hyperparameter. To obtain ϵ, they use Projected Gradient Descent [27]. A qualitative analysis of the feature embeddings of bias-aligned, bias-conflicting and adversarial samples made with t-SNE showed that although the changes made to obtained adversarial samples seem negligible from the human eye, they are significant from the network perspective, and therefore enable a large increase in performance (cf section IV-C for results).

2) Post-processing methods: Post-processing have the advantage over pre-processing and in-processing of not requiring to retrain models or to gather additional data. The ability not to retrain a model is a huge benefit with the advent of large foundation models expensive to train. However, to the best of our knowledge, no pure debiasing methods have yet proposed a solution, although methods from Fairness exists. Adapting these methods to debiasing is impossible in practice, because it requires to know the true correlation between B^* and Y^*, which is supposed unknown. We will therefore focus on the rich literature of in-processing methods.

3) In-processing methods:

a) Prior-guided: Prior-guided or weakly supervised methods do not assume access to bias labels, but they assume prior knowledge about the bias. In ReBias [6], the assumption is that the bias is the texture (*i.e.*, the models use the local texture of the objects rather than their shape to classify them). Therefore, BA models are designed by using convolutional neural networks with small receptive fields with 1x1 kernels. A debiased model is then trained by forcing its output to be independent to the output of the BA[13].

However, presuming the type of the bias in advance limit the applicability of these methods. Another downside of prior-guided approaches is that they require prior knowledge that can be expensive to acquire.

b) Invariant risk minimization: One approach to bias mitigation is to consider that, since bias features are only spuriously correlated with the target variable, and therefore are not causal, finding the causal features of the data ought to get rid of the biased ones. In their paper, Arjovsky *et al.* [5] aim at discovering those causal features by finding invariant ones. They consider a set of environments $\mathcal{S}$ from which the data can come from, and state that a representation of the data $\mathcal{E} : \mathcal{X} \longrightarrow \mathcal{H}$ elicits an invariant predictor $f \triangleq \mathcal{C} \circ \mathcal{E}$ across environments $\mathcal{S}$ if there exists a classifier $\mathcal{C} : \mathcal{H} \longrightarrow \mathcal{Y}$ simultaneously optimal for all environments, *i.e.*, $\mathcal{C} \in \underset{\overline{\mathcal{C}}:\mathcal{H}\longrightarrow\mathcal{X}}{\arg\min} R^e(\overline{\mathcal{C}} \circ \Phi)$, where $R^e(f) = \mathbb{E}[l(f(X), Y)|E = e]$ is the risk under environment $e \in \mathcal{S}$. A data representation that elicits a non-trivial (hence useful) invariant predictor is a solution of the following constrained optimization problem:

$$\begin{aligned} \min_{\mathcal{E}:\mathcal{X}\longrightarrow\mathcal{H},\ \mathcal{C}:\mathcal{H}\longrightarrow\mathcal{Y}} \quad & \sum_{e\in\mathcal{S}_{train}} R^e(\mathcal{C} \circ \mathcal{E}) \\ \text{subject to} \quad & \mathcal{C} \in \underset{\overline{\mathcal{C}}:\mathcal{H}\longrightarrow\mathcal{X}}{\operatorname{argmin}} \ R^e(\overline{\mathcal{C}} \circ \mathcal{E}), \quad \forall e \in \mathcal{S}_{train} \end{aligned} \tag{IRM}$$

which is relaxed into the tractable version:

$$\min_{\mathcal{C}:\mathcal{X}\longrightarrow\mathcal{H}} \sum_{e\in\mathcal{S}_{tr}} R^e(\mathcal{C}\circ\mathcal{E}) + \lambda \left\| \nabla_{\mathcal{C}|\mathcal{C}=1.0} R^e(\mathcal{C} \circ \mathcal{E}) \right\|^2 \tag{IRMv1}$$

Although minimizing IRMv1 works in some settings, further works [23] have noted that the performance of IRMv1 deteriorated to or below the level of an ERM[14] model when the model was large or when the amount of data was small.

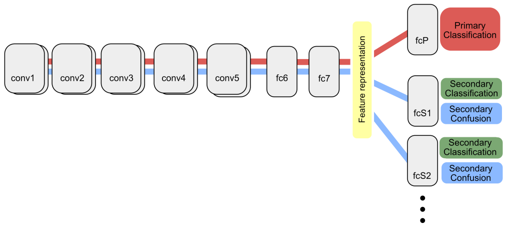

Fig. 6. Overview of the model architecture in BlindEye (from [4]).

c) Learning unbiased representations: Another solution to learn an unbiased classifier is to ensure that the feature representations are unbiased, *i.e.*, no bias information can be retrieved from the bottleneck layer.

[13] We do not present the maths for this reference, because they are quite complicated and not necessary as simpler methods outperform Rebias.

[14] ERM: Empirical Risk Minimization. An ERM model is a model trained with the standard methods of deep learning, without any debiasing methods.

Perhaps the simplest approach is to seek to learn feature representation without any indication of class, with self-supervised learning. In [18], they train the backbone of both BA and debiased model with self-supervised learning via the BYOL objective. They show that a simple linear ERM classifier on top of a backbone trained with self-supervised learning outperforms by a large margin an ERM model trained in an end-to-end fashion.

Alvi *et al.* [4] (BlindEye) use a joint learning and unlearning loss to learn unbiased representations. At the end of the backbone of the network, they attach a primary classification head, tasked to classify the input (cf Fig.6). They also attach one more head per bias attribute b, each bias removal head is composed of a classifier, trying to infer the bias from the feature representation, and a confusion loss, measuring how much the classifier is confused (*i.e.*, how close the prediction of the bias is from a uniform distribution). At each epoch, we first train the heads classifying the bias attributes by minimizing a classification loss (*e.g.* cross-entropy) and then we train the backbone and the primary classification head by minimizing the joint loss:

$$\begin{aligned}\mathcal{L}(x, y, b; \theta_{repr}, \theta_p, \theta_b) &\triangleq \\ &\mathcal{L}_p(x, y; \theta_{repr}, \theta_p) + \alpha \mathcal{L}_{conf}(x, b; \theta_{repr}, \theta_b)\end{aligned}$$

where $\theta_{repr}, \theta_p, \theta_b$ are the parameters of the backbone, the primary classification head and the bias classification heads respectively, and where $\mathcal{L}_p(x, y; \theta_{repr}, \theta_p)$ is the classification loss for the primary task, α is a hyperparameter that determines how strongly the confusion loss affects the joint loss and:

$$\mathcal{L}_{conf}(x, b; \theta_{repr}, \theta_b) \triangleq \sum_{i \in \mathcal{B}} \beta_i \mathcal{L}_{conf,i}(x, b; \theta_{repr}, \theta_{b_i})$$

where β_i is a weight assigned to bias attribute i and $\mathcal{L}_{conf,i}(x, b; \theta_{repr}, \theta_{b_i})$ is the cross-entropy between the output of the classifier for the bias i and a uniform distribution. The authors remark that although the method allows a significant accuracy improvement in the extremely biased settings, their is no similar improvement or even no improvement at all of the accuracy in less biased settings.

FairKL [7] is another method to learn unbiased representations. The authors propose a new supervised contrastive loss, aiming to pull the representations of samples from the same class together, while making sure that the bias-conflicting and bias-aligned samples behave in a similar way. Formally, given an "anchor" sample $x \in \mathcal{X}$, we denote samples from the same class as x_i^+ ("positive" samples) and samples from other classes as x_i^- ("negative" samples). In contrastive learning, we seek a parametric mapping $f : \mathcal{X} \to \mathbb{S}^{d-1}$ such that similar samples are close to each other in the representation space. We define a similarity (resp. dissimilarity) measure s (resp. d) on $\mathbb{S}^{d-1}$. The authors want to ensure that the positive samples are mapped close to $f(x)$ and that negative samples are mapped far away from $f(x)$ so that there is a margin of size at least ϵ between the positive representations and the negative ones :

$$\forall i, j, \quad s_j^- - s_i^+ \leq -\epsilon \tag{1}$$

where $s_j^- = s(f(x), f(x_j^-))$ and $s_i^+ = s(f(x), f(x^+))$. They derive the ϵ-SupInfoNCE loss:

$$\begin{aligned}&\sum_i \max\left(-\epsilon, \left\{s_j^- - s_i^+\right\}_{j=1,\ldots,N}\right) \\ &\qquad\approx -\sum_i \log\left(\frac{\exp(s_i^+)}{\exp(s_i^+ - \epsilon) + \sum_j \exp(s_j^-)}\right)\end{aligned} \tag{ϵ-SupInfoNCE}$$

where the left hand side is the direct translation of the constraints, which is not differentiable, and the right hand side is the smoothed loss. This loss only ensures that the representations of the samples from different classes are distinct, but it does not take into account the bias. We will note bias-aligned (resp. conflicting) samples with $x^{\cdot,b}$ (resp. $x^{\cdot,b'}$). While satisfying the contrastive constraints (1), there could be a bias in the representations in the case where the following ordering happens:

$$\forall i, j, k, l \qquad d_i^{+,b} < d_j^{+,b'} < d_k^{-,b} - \epsilon < d_l^{-,b'} - \epsilon \tag{2}$$

where

$$\begin{aligned}d_i^{+,b} &= d(f(x), f(x_i^+, b))) \\ d_j^{+,b'} &= d(f(x), f(x_j^{+,b'})) \\ d_k^{-,b} &= d(f(x), f(x_k^{-,b})) \\ d_l^{-,b'} &= d(f(x), f(x_l^{-,b'}))\end{aligned}$$

To prevent that from happening, the authors define a new set of debiasing regularizers that ensure that the distributions $\{d_k^{\cdot,b}\}$ and $\{d_i^{\cdot,b'}\}$ are similar (*e.g.* assume that the distributions are gaussians and set the regularizer as the KL divergence between them). The combination of the contrastive loss and the debiasing regularization term gives better performances than the previously proposed debiasing contrastive losses.

d) Mutual information minimization.: Similarly to the approach of Alvi *et al.* [6], Nahon *et al.*[28] try to minimize the amount of information about the bias that is contained in the bottleneck layer of the neural network being debiased. They follow the technique of IRENE[15] [37]: in addition to the classification head, an information removal head (IRH) is plugged to the bottleneck layer. The IRH is trained to predict a pseudo-bias distribution $\hat{b}_n$ for each sample x_n by minimizing the cross entropy loss $\mathcal{L}(b_n, \hat{b}_n)$. Then we can compute the mutual information between B and $\hat{B}$ under the bias conflicting condition as follows:

$$\mathcal{I}^\perp = \sum_{j,k \in \mathcal{B}^2} p_{B,\hat{B}}^\perp(j,k) \log\left(\frac{p_{B,\hat{B}}^\perp(j,k)}{p_B^\perp(j) \cdot p_{\hat{B}}^\perp(k)}\right)$$

where

$$p_{B,\hat{B}}^\perp(j,k) = \frac{1}{|\mathcal{D}^\perp|} \sum_{n \in \mathcal{N}_{\mathcal{D}^\perp}} \delta_{j,b_n} \sigma(\hat{b}_n)_k$$

where $\mathcal{D}^\perp$ is the set of bias-conflicting samples. $\mathcal{I}^\perp$ plays the role of a loss, and is then backpropagated directly through

[15] acronym: Information REmoval at the bottleNEck

the backbone starting from the bottleneck layer. It ensures that the representation of bias-conflicting samples in the latent space does not contain information that is related to the bias attribute The mutual information is not computed on bias-aligned samples because it would destroy the information needed for classification.

e) Reweighting.: Many recent works have studied reweighting methods. Following [22], we can formulate reweighting techniques in a unified framework as methods that minimize the reweighted empirical risk $\mathcal{R}_w(\theta)$ defined as follows:

$$\hat{\mathcal{R}}_w(\theta) = \mathbb{E}_{(x,y)\in\mathcal{D}}\left[\mathcal{W}(x,y;\theta,\phi)\cdot\mathcal{L}(x,y;\theta))\right]$$

In JTT [24], the authors use the hyperparameter λ_{up} to up-weight the bias-conflicting samples identified thanks to the BA g_ϕ:

$$\mathcal{W}(x,y;\theta,\phi) = \lambda_{\text{up}}(1-\delta_{b_\phi,y}) + 1\cdot\delta_{b_\phi,y}$$

where $\delta_{b_\phi,y} = 1$ if x is bias-aligned and 0 otherwise.

In LfF [30], the authors define the reweighting factor as a relative difficulty score :

$$\mathcal{W}(x,y;\theta,\phi) = \frac{\text{CE}(g_\phi(x),y)}{\text{CE}(f_\theta(x),y)+\text{CE}(g_\phi(x),y)} \tag{3}$$

The intuition being that for bias-conflicting samples, the biased model is expected to have larger cross-entropy (CE) loss compared to the debiased model, resulting in a large weight for the training of the debiased model.

In VCBA [28], the authors first use the BA model g_ϕ to infer the bias labels $\mathcal{B}_\phi = (b_{\phi,n})_{n\in\mathcal{N}_\mathcal{D}}$, assuming an underlying ground truth $\hat{\mathcal{B}} = (\hat{b}_n)_{n\in\mathcal{N}_\mathcal{D}}$. The loss is then reweighted as follows:

$$\mathcal{W}(x_n,y_n;\theta,\phi) = \frac{1}{\rho_{b_{\phi,n}}}\cdot\delta_{b_{\phi,n},y_n} + \frac{1}{1-\rho_{b_{\phi,n}}}\cdot\overline{\delta}_{b_{\phi,n},y_n}$$

where we define

$$\rho_y = \frac{\left|\mathcal{D}^{||}_{\phi,y}\right|}{\left|\mathcal{D}^{||}_{\phi,y}\bigcup\mathcal{D}^{\perp}_{\phi,y}\right|}$$

where $\mathcal{D}^{||}_{\phi,y}$ is the subset of correctly classified samples for the class y and $\mathcal{D}^{\perp}_{\phi,y}$ the subset of misclassified samples for the class y. Intuitively, the more bias-aligned samples there are, the more the bias-conflicting samples are up-weighted.

f) Distributionally robust optimization: In GroupDRO [36], the authors build upon the literature in distributionally robust optimization (DRO), they define groups $g\in\mathcal{G}$ by grouping together all the samples having the same target class and bias attribute $(y,b)\in\mathcal{Y}\times\mathcal{B}$. Then, they maximize the worst group performance by minimizing the empirical worst-group risk:

$$\hat{\mathcal{R}}(\theta) = \max_{g\in\mathcal{G}}\mathbb{E}_{(x,y)\in g}\left[\mathcal{L}(x,y;\theta)\right]$$

This yields a model with good worst-group *training* loss. However, it does not imply good worst-group *test* loss because of the generalization gap $\delta(\theta) = \mathcal{R}(\theta) - \hat{\mathcal{R}}(\theta)$ (where $\mathcal{R}(\theta)$ is the expected worst group loss on the true underlying distribution, estimated on the test set). Indeed, for over-parameterized model, both ERM and GroupDRO models achieve high average training accuracy *and* high worst-group training accuracy and generalize well on average but perform really poorly in terms of worst-group test accuracy. By applying strong regularization (such as a high coefficient for ℓ_2 penalty), the authors prevent the model to achieve perfect training accuracy but allow them to generalize better. In particular, GroupDRO models improve drastically their worst group test accuracy. Interestingly, the authors also prove that for non-convex loss, GroupDRO models will always have equivalent of better worst-group test accuracy than models obtained by reweighting the loss.

g) Feature augmentation: One shortfall of reweighting techniques, pointed out by [21] and showed experimentally in [21], is that when the number of bias-conflicting samples is really low, the methods often fail because they tend to overfit on the small subset of bias-conflicting samples and therefore fail to learn generalizable representation. This has lead to the development of methods aiming to increase synthetically the number of bias-conflicting samples. In section IV-A, we presented BiasAdv [22], which performs dataset augmentation. Before BiasAdv, the authors of [21] introduced Disentangled Feature Augmentation (DFA), a method to augment the latent space features. They train two separate encoders E_i and E_b and their classifiers C_i and C_b. Encoders E_i and E_b embed an image $x\in\mathcal{X}$ into intrinsic feature vectors $z_i = E_i(x)$ and bias feature vectors $z_b = E_b(x)$, respectively. Afterward, linear classifiers C_i and C_b take the concatenated vector $z = [z_i; z_b]$ as input to predict the target label y. The loss $\mathcal{L}_{dis}$ is used to enforce disentanglement:

$$\mathcal{L}_{dis} = W(z)CE(C_i(z),y) + \lambda_{dis}GCE(C_b(z),y)$$

where $W(z) = \frac{CE(C_b(z),y)}{CE(C_i(z),y)+CE(C_b(z),y)}$ is the relative difficulty score (based on eq (3) from LfF [30]) and CE and GCE are used to push disentanglement, following [30]. The loss from C_i is backpropagated only to E_i while the loss from C_b is backpropagated only to E_b. After a warm-up phase, the features are starting to be disentangled, so we can add another loss $\mathcal{L}_{swap}$. To increase the diversity of bias-conflicting samples at the feature level, the features z_b are randomly swapped between samples from the same mini-batch; we obtain $z_{swap} = [z_i; \tilde{z}_b]$ where $\tilde{z}_b$ are the bias features of another sample in the mini-batch, with class label $\tilde{y}$:

$$\mathcal{L}_{swap} = W(z)CE(C_i(z_{swap},y) + \lambda_{swap}GCE(C_b(z_{swap},\tilde{y}))$$

Figure 7 illustrates the effective disentanglement of bias and intrinsic features. The downside of this approach is that it does not work as well on real-world datasets [22], because the latent features are both more numerous and entangled.

C. Comparison

The behavior of DNNs is difficult to study in theory because of the many non-linearity in their architecture. Therefore,

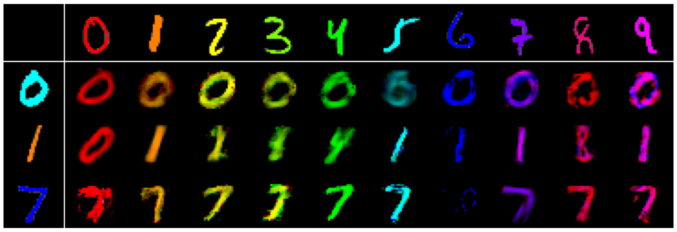

Fig. 7. Reconstructed images from disentangled representation in Colored MNIST. The first column (resp. the first row) indicate the samples where the bias attribute (color) (resp. the intrinsic attribute (digit)) are extracted. Upon swapping the bias attributes, we observe that the color changes while maintaining the digit (taken from [21]).

the usual protocol to compare methods in Deep Learning and therefore in Debiasing is to apply the methods to the same benchmark dataset while keeping the variations between the experiments as small as possible. Then the methods are compared through performance metrics, such as top-1 accuracy, average group accuracy or worst group accuracy. Top-1 accuracy (or simply accuracy) is equal to the proportion of test samples on which the model predicted the correct target label. The concept of "group" is prevalent in DRO. In the context of debiasing, there is one group per pair $(y, b) \in \mathcal{Y} \times \mathcal{B}$ (*e.g.* (blond,male), (blond,female), (not blond, male), (not blond, female) for the CelebA dataset).

In Table I, we summarized the performance metrics of the aforementioned debiasing methods on a few benchmark datasets. We first observe that none of the methods are evaluated on many datasets. Moreover, the values reported by the papers are not always compatible. For instance, [24], [10] and [36] report different worst group accuracy for a vanilla ERM model on the Waterbirds dataset. It can be explained by differences in the pre-processing of the dataset, in the architecture of the model or in the hyperparameter used. In the table II in Appendix VII-B, we reported all the values. In Table I, in case of conflict, we reported the value from the original paper presenting the method, and we reported the highest value when the Vanilla ERM model was concerned. We can also note that some papers [7, 30, 28, 21] run their experiment 3 times with different seeds, and then report the average and the standard deviation of the performance metric, whereas other report a single value [24, 6, 10, 24], which makes the interpretation harder.

From the results, we can see that among the presented methods the best performing one until now is the combination of LfF and the adversarial augmentation proposed in BiasAdv [22]. Interestingly, it is an unsupervised method, where bias information is supposed unavailable. Next is the contrastive framework for learning unbiased representation FairKL [7]. It is difficult to infer much more from these results alone because the methods are not evaluated on the same datasets, which is a serious flaw in the literature. Another issue is that most of the methods are presented as tightly packed systems, and it is difficult to see what part of the method can be changed without breaking it. As a result, no systematic review have yet evaluated the effect of each of the individual components of the methods. For instance, it would have been interesting to be able to see the results of a method with the following pipeline : use LWBC [18] to find samples that are almost surely bias aligned, train a BA on the bias-aligned samples using VCBA [28], use BiasAdv to augment the dataset with synthetic bias-conflicting samples and then train a debiased model using FairKL [7].

V. Biasedness metrics

In this section, the notions of bias and of a biased classifier we use are more nuanced: we drop the hypothesis (WH2) (*i.e.*, that $B^* \perp Y^*$). Now only (WH1) holds. In other words we consider that a dataset or a model are biased if the correlation between the bias attribute and the target attribute change between the underlying true distribution and the distributions defined by a dataset or a model.

Biasedness metrics are useful for two reasons. First they are a tool to compare bias-extraction methods. In fact, given the prevalence of debiasing methods requiring a biased model (BA: *Bias Amplifier*), we could expect that the authors provide metrics indicating how well the BA captures the bias, but that is rarely the case. Often [24], the authors declare that the BA was indeed biased because the debiased model learned downstream performs better than a vanilla ERM model. However, this does not allow us to disentangle the respective importance of the biasedness of the BA, the efficacy of the bias extraction method and the efficacy of the bias mitigation methods. Biasedness metrics are also used in methods to intentionally train a BA. Following the method introduced by Nahon *et al.* [28], if one is able to define a biasedness metric that only relies on the output of the model and not on the ground truth bias information, then it is possible to use it to detect during training when a model is the most biased, and thus to use it as BA.

Although the field of debiasing is paradoxically poor in biasedness metrics, the literature of fairness and distributionally robust optimization (DRO) have defined and analyzed numerous metrics, although none has achieved consensus within the community as the definitive standard. We will present the more commonly used biasedness metrics. The goal is not to provide an exhaustive list of existing metrics, but rather to highlight the various approaches, how they bring different aspects of bias to light and how they can be incompatible. Finally we will see a recent trend for bias discovery.

A. Accuracy-based metrics

Nahon *et al.* [28] use the **bias-conflicting accuracy** to assess the biasedness of the model :

$$\zeta = \frac{1}{|\mathcal{D}^{\perp}|} \sum_{(x,y) \in \mathcal{D}^{\perp}} \delta_{f_\theta(x), y} \in [0, 1]$$

where $\mathcal{D}^{\perp}$ is the set of all bias-conflicting samples and $f_\theta(x)$ denotes the class predicted for x by the model. This metric is ambiguous, because even though $\zeta \approx 1$ indicates that the model is not biased, $\zeta \approx 0$ does not necessarily imply that the model uses the bias to classify the samples. Indeed, if the

TABLE I

RESULTS OF THE PRESENTED METHODS ON A SELECTION OF DATASETS. BEST PERFORMING RESULTS ARE MARKED IN BOLD, SECOND BEST ARE UNDERLINED. BEST UNSUPERVISED RESULTS ARE HIGHLIGHTED.

Benchmark	Method	Vanilla ERM	**ϵ-SupInfoNCE +FairKL**[7]	**LfF** [30]	**GroupDRO** [36]	**VCBA** [28]	**ReBias** [6]	**DFA** [21]	**EIIL**[10] [10]	**JTT** [24]	**LfF+BiasAdv** [22]
	Type of method	Unsupervised	Supervised	Unsupervised	Supervised	Unsupervised	Prior guided	Unsupervised	Unsupervised	Weakly Supervised	Unsupervised
	Variant										
Biased MNIST	$\rho^{a} = 0.99$	90.8 ± 0.3	$\mathbf{97.86 \pm 0.02}$	95.1	-	97.7 ± 0.3	88.1	-	-	-	-
(unbiased)[b]	$\rho = 0.999$	11.8 ± 0.7	$\mathbf{90.51 \pm 1.55}$	15.3	-	58.7 ± 21.8	22.7	-	-	-	-
Colored MNIST	$\rho = 0.95$	82.17 ± 0.74	-	85.39 ± 0.94	84.50 ± 0.46	-	$\mathbf{96.96 \pm 0.04}$	89.66 ± 1.09	-	-	-
(unbiased)	$\rho = 0.995$	35.34 ± 0.13	-	63.39 ± 1.97	59.67 ± 2.73	-	$\mathbf{70.47 \pm 1.84}$	65.22 ± 4.41	-	-	-
Waterbirds	Avg group acc	97.3	-	**97.5**	93.5 ± 0.3	-	-	-	96.9	93.6	-
	Worst group acc	72.6	-	75.2	$\mathbf{91.4 \pm 1.1}$	-	-	-	78.7	86.0	-
CelebA (target="Blond Hair")	unbiased	79.0	-	84.24 ± 0.37	85.43 ± 0.53	$\mathbf{90.2 \pm 1.1}$	-	-	-	-	-
(bias="Gender")	bias-conflicting[c]	59.0		81.24 ± 1.38	83.40 ± 0.67	$\mathbf{84.5 \pm 2.0}$	-	-	-	-	-
CelebA (target="Heavy makeup")	unbiased	62.00 ± 0.02	-	$\mathbf{66.20 \pm 1.21}$	64.88 ± 0.42	-	-	-	-	-	-
(bias="Gender")	bias-conflicting	33.75 ± 0.28	-	45.48 ± 4.33	$\mathbf{50.24 \pm 0.68}$	-	-	-	-	-	-
Corrupted CIFAR-10^{0}	$\rho = 0.95$	39.42 ± 0.64	50.73 ± 0.90	50.72 ± 1.31	-	-	43.43 ± 0.41	51.13 ± 1.28	-	41.20 ± 0.19	$\mathbf{57.78 \pm 0.33}$
(unbiased)	$\rho = 0.995$	23.08 ± 1.25	33.33 ± 0.38	28.81 ± 0.44	-	-	22.27 ± 0.41	29.95 ± 0.71	-	23.66 ± 0.78	$\mathbf{36.78 \pm 0.20}$
9-class ImageNet	biased set	94.0 ± 0.1	95.1 ± 0.1	91.2 ± 0.1	-	$\mathbf{95.5 \pm 0.2}$	94.0 ± 0.2	-	-	-	-
	unbiased set	92.7 ± 0.2	$\mathbf{94.8 \pm 0.3}$	89.6 ± 0.3	-	-	92.7 ± 0.2	-	-	-	-
ImageNet-A	-	30.5 ± 0.5	$\mathbf{35.7 \pm 0.5}$	29.4 ± 0.8	-	34.2 ± 0.9	30.5 ± 0.2	-	-	-	-
Multicolor MNIST	Unbiased	57.2	-	52.0	-	$\mathbf{73.1 \pm 0.9}$	-	-	69.7	-	-
BAR	$\rho = 0.95$	68.60 ± 2.25	-	67.48 ± 0.46	-	-	65.00	63.5	-	68.53 ± 3.29	$\mathbf{72.62 \pm 0.11}$
	$\rho = 0.99$	57.65 ± 2.36	-	57.71 ± 3.12	-	-	52.1	52.3	-	58.17 ± 3.30	$\mathbf{63.20 \pm 2.64}$
	$\rho = 1.0$	51.85 ± 5.92	-	$\mathbf{62.98 \pm 2.76}$	-	-	59.74 ± 1.49	-	-	-	

[a] ρ always indicates the proportion of bias-aligned samples in the training set.
[b] "unbiased" indicates that the test set (on which the model was evaluated) is unbiased, *i.e.*, the groups $(y, b) \in \mathcal{Y} \times \mathcal{B}$ are of the same size.
[c] "bias-conflicting" indicates that the test set (on which the model was evaluated) is composed solely of bias-conflicting samples.

model misclassifies *all* the samples then $\zeta = 0$, but the model did not learn biased features.

Nam *et al.* [30] and Ciranni *et al.* [9] use a more complete approach, by studying both **the bias-conflicting accuracy and the bias-aligned accuracy**. The model is deemed biased when the bias-conflicting accuracy is low and the bias-aligned accuracy is high. Indeed, it tells us that the majority of bias-conflicting samples are misclassified and that the majority of the bias-aligned samples are correctly classified. The problem is that these metrics are both coarse and difficult to interpret.

In the Distributionally robust optimization (DRO) literature, the metrics focus on the output disparity across subgroups. The notion of subgroups can be easily transposed to the context of debiasing by considering the subset of samples sharing the same target class and bias information as one subgroup. We therefore have one subgroup per pair $(y, b) \in \mathcal{Y} \times \mathcal{B}$. The two main metrics are **worst group accuracy (WG)** [36, 3, 41] and the **gap between average and worst group accuracy (GAP)** [3]. These metrics allow to detect as biased a classifier which could have good bias-aligned accuracy and bias-conflicting accuracy while still performing poorly on a specific subgroup.

B. Fairness metrics

The Fairness literature provides us with many metrics to try to describe the fairness of a model's decisions. They can also be applied to measure the bias, even though they must often be adapted to a multi-class setting. We will only focus on two of the most used metrics : Demographic Parity (DP) and Equalized Odds (EO).

The objective of DP [3, 2, 38] is to measure how much the 'independence' between the target attribute and the bias attribute is violated. In the context of binary classification, DP measures the difference in mean outcomes across groups (here, a group is defined as all samples sharing the same bias information). The authors of [2] extend the definition to the multi-class setting as follows:

$$\mathrm{DP} = \max_{y \in \mathcal{Y}} \left(\max_{b \in \mathcal{B}} \mathbb{E}\left[f_y(X) | B = b\right] - \min_{b \in \mathcal{B}} \mathbb{E}\left[f_y(X) | B = b\right] \right) \tag{DP}$$

where $f_y(x)$ denotes the model's estimate of the probability that the sample x belongs to class y. Models with small DP have similar mean probability across all groups with respects to all classes. Although DP is straightforward and seems to capture the intuitive idea of fairness, Hardt *et al.* [12] point out that in the realistic case where the target attribute and the bias attribute are correlated, trying to enforce DP would result in a degradation of the performances of the model (in particular, the perfect classifier would not be allowed). Hence, they propose **Equalized Odds (EO)**. A given predictor model is said to satisfy Equalized Odds if $\hat{Y}$ and B are independent conditional on Y [12]. It is transformed into the following multi-class metric by [2]:

$$\begin{aligned} \mathrm{EO}(f_\theta) = \max_{y \in \mathcal{Y}} [& \max_{(y', b) \in \mathcal{Y} \times \mathcal{B}} \mathbb{E}\left[f_y(X) | Y = y', B = b\right] \\ & - \min_{(y', b) \in \mathcal{Y} \times \mathcal{B}} \mathbb{E}\left[f_y(X) | Y = y', B = b\right]] \end{aligned} \tag{EO}$$

which has the huge advantage of accepting the perfect classifier as a solution. As for the intuition, in the binary case, EO ensures that the true positive rate and false positive rate are the same across all the classes.

C. Estimating biasedness by characterizing the mistakes

The authors of [3] present SKEWSIZE, a metric that is more specific than accuracy-based and fairness metrics in that it can pinpoint a link between a bias attribute and a target class. The idea is to measure to what extent the predictions of the model on samples of a given class $y' \in \mathcal{Y}$ are affected by a given bias attribute $b' \in \mathcal{B}$. Considering that for each $x \in \mathcal{X}$ the parametric model f_θ defines a conditional distribution $q(Y|X = x; \theta)$, the idea of measuring the distributional bias that b' has on class y' is to compare the induced family of distributions defined by $f_\theta(x)$ when $x \sim p(X|Y = y', B = b')$ and $x \sim p(X|Y = y', B \neq b')$:

$$\mathcal{H}(Q_A(Y|X;\theta)||Q_B(Y|X,\theta))$$

where $Q_A(Y|X;\theta)$ (resp $Q_B(Y|X,\theta)$)denotes the family of distributions obtained when $B = b'$ (resp when $B \neq b'$), ie $Q_A(Y|X;\theta) = (q\,(Y|X = x;\theta))_{x\sim p(X|Y=y',B=b')}$ and $\mathcal{H}(.||.)$ is an operator that accounts the similarity between those distributions. The authors focus on classification tasks, where they actually compare Q_A and Q_B by estimating the effect size via the Cramér's V statistic defined as:

$$\nu_{y'} = \sqrt{\frac{\chi^2_{y'}}{N_{y'} \cdot DF}} \tag{4}$$

where N is the sample size, DF is the number of degrees of freedom and χ^2 is the test statistic from the Pearson's chi-squared independence test (cf Appendix VII-A).

In order to be able to compare models, we need a metric which is not class-specific (as above) and which can be summarized in a single scalar. The metric should indicate a low bias when only a few classes have a high estimated effect size and it should indicate a high bias when the predictions on a large proportion of the classes are highly dependent on the bias. The authors propose to use the Fisher-Pearson coefficient of skewness on the estimated effect sizes $\{\nu_y\}_{y\in\mathcal{Y}}$ with empirical mean $\overline{\nu}$:

$$\textsc{SkewSize} = \frac{\sum_{y\in\mathcal{Y}}(\nu_y - \overline{\nu})^3}{\left(\sum_{y\in\mathcal{Y}}(\nu_y - \overline{\nu})^2\right)^{3/2}}$$

SKEWSIZE is capable of surfacing biases not captured by other metrics because it not only considers the change of distribution across subgroups on the correct class but also on the other classes.

D. Bias discovery

One recent trend in debiasing is using vision-language models to extract human-interpretable descriptions of potential biases affecting a deep image classification model. In [9], the authors present Say My Name (SaMyNa), a bias discovery framework. It works in 5 steps. Before anything, fix a target class $y \in \mathcal{Y}$. First a subset of representative samples is extracted from the cluster formed by samples classified as y by the model under study. Second, a Vision-Language model captions all selected images. Third, the most recurrent keywords from the captions are selected. Fourth, a class embedding is extracted by grouping the common features of the generated captions from the latent space of a pre-trained text encoder. Finally, the keywords are ranked by cosine similarity with the class embedding. This framework allows for the identification in natural language of potential dataset biases, starting from an unlabeled dataset.

VI. CONCLUSION: FUTURE RESEARCH DIRECTIONS

a) Class-specific bias extraction: We have seen in section IV-A that one technique [28] used to intentionally train a BA is to train a model on a biased dataset and to find the epoch during training when the model uses the most the bias to make its predictions. However, in real scenarios, the model could learn multiple biases at different epochs during training. Hence, it could be beneficial not to chose a single epoch during training where the model exhibits a biased behavior, but rather several epochs, one per class for instance, because biases attributes are typically associated with a specific class.

b) Elucidating the design space of debiasing methods: Following the same thought process as in [17], one can observe that the literature of debiasing is full of contributions presenting end-to-end methods expressed in their own theoretical framework. However, this approach has a danger of obscuring the available design space — a proposed method may appear as a tightly coupled package where no individual component can be modified without breaking the entire system[16]. Hence, one research direction could be to systematically evaluate how well the existing bias-extraction methods combine with bias-mitigation methods, in order to assess the actual best performing methods and to uncover the most promising research directions.

c) Improving the extraction of the bias information from the BA: BAs are often used to infer pseudo bias labels for the training samples. It is also the only way to extract the bias information from the BAs so that it can then be used by supervised methods. The most common heuristic is to use the BAs to predict the classes of the training samples and then to assign the pseudo bias label "bias-conflicting" to the samples that were misclassified by the BAs and "bias-aligned" to the others [28, 24]. This heuristic seems overly simple and none of the reviewed papers present an evaluation of how well the pseudo bias labels match the ground truth. One research direction is to evaluate how well this heuristic perform and to try to define another, finer, method.

d) General debiasing: The problem of debiasing in general where we do not have hypothesis (WH2) remains to be solved. One final research direction that could be explored is to search how to get rid of the assumption $Y^* \perp B^*$.

[16] This sentence is used as such in [17] but applies as well to debiasing.

References

[1] Ibrahim Alabdulmohsin and Mario Lucic. “A Near-Optimal Algorithm for Debiasing Trained Machine Learning Models”. In: *Advances in Neural Information Processing Systems*. 2021.

[2] Ibrahim M Alabdulmohsin, Jessica Schrouff, and Sanmi Koyejo. “A Reduction to Binary Approach for Debiasing Multiclass Datasets”. In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 2480–2493.

[3] Isabela Albuquerque et al. “Evaluating Model Bias Requires Characterizing its Mistakes”. In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. Proceedings of Machine Learning Research. July 2024, pp. 938–954.

[4] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellaaker. “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) Workshops*. Sept. 2018.

[5] Martin Arjovsky et al. “Invariant risk minimization”. In: *arXiv preprint arXiv:1907.02893* (2019).

[6] Hyojin Bahng et al. “Learning De-biased Representations with Biased Representations”. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. July 2020, pp. 528–539.

[7] Carlo Alberto Barbano et al. “Unbiased Supervised Contrastive Learning”. In: *ICLR*. May 2023.

[8] Anu Bradford. *The Brussels Effect: How the European Union Rules the World*. Oxford University Press, Feb. 2020. ISBN: 9780190088583.

[9] Massimiliano Ciranni et al. *Say My Name: a Model's Bias Discovery Framework*. 2024. arXiv: 2408.09570 [cs.LG].

[10] Elliot Creager, Joern-Henrik Jacobsen, and Richard Zemel. “Environment Inference for Invariant Learning”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 2189–2200.

[11] Suchin Gururangan et al. “Annotation Artifacts in Natural Language Inference Data”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 2018, pp. 107–112.

[12] Moritz Hardt et al. “Equality of Opportunity in Supervised Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016.

[13] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[14] Dan Hendrycks and Thomas Dietterich. “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”. In: *International Conference on Learning Representations*. 2019.

[15] Dan Hendrycks et al. “Natural Adversarial Examples”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 15257–15266.

[16] Youngkyu Hong and Eunho Yang. “Unbiased Classification through Bias-Contrastive and Bias-Balanced Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 26449–26461.

[17] Tero Karras et al. “Elucidating the Design Space of Diffusion-Based Generative Models”. In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 26565–26577.

[18] Nayeong Kim et al. “Learning Debiased Classifier with Biased Committee”. In: *Advances in Neural Information Processing Systems*. 2022.

[19] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[20] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[21] Jungsoo Lee et al. “Learning Debiased Representation via Disentangled Feature Augmentation”. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 25123–25133.

[22] Jongin Lim et al. “BiasAdv: Bias-Adversarial Augmentation for Model Debiasing”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 3832–3841.

[23] Yong Lin et al. “Bayesian Invariant Risk Minimization”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 16000–16009.

[24] Evan Z Liu et al. “Just Train Twice: Improving Group Robustness without Training Group Information”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. July 2021, pp. 6781–6792.

[25] Shuying Liu and Weihong Deng. “Very deep convolutional neural network based image classification using small training sample size”. In: *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. 2015, pp. 730–734.

[26] Ziwei Liu et al. “Deep Learning Face Attributes in the Wild”. In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[27] Aleksander Madry et al. “Towards Deep Learning Models Resistant to Adversarial Attacks”. In: *International Conference on Learning Representations*. 2018.

[28] 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 (ICCV)*. Oct. 2023.

[29] Rémi Nahon et al. “Debiasing Surgeon: Fantastic Weights and How to Find Them”. In: *Computer Vision – ECCV 2024: 18th European Conference, Milan,*

Italy, September 29–October 4, 2024, Proceedings, Part LXXXV. 2024, pp. 435–452.

[30] Junhyun Nam et al. “Learning from failure: De-biasing classifier from biased classifier”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 20673–20684.

[31] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. “Discrimination-aware data mining”. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2008, pp. 560–568.

[32] Jiaxin Qi et al. “Class Is Invariant to Context and Vice Versa: On Learning Invariance for Out-Of-Distribution Generalization”. In: *Computer Vision – ECCV 2022*. 2022, pp. 92–109. ISBN: 978-3-031-19806-9.

[33] “Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)”. In: *Official Journal of the European Union* L 119 (2016), pp. 1–88.

[34] “Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance)”. In: *Official Journal of the European Union* (2024).

[35] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.

[36] Shiori Sagawa et al. “Distributionally Robust Neural Networks”. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020.

[37] Enzo Tartaglione. “Information Removal at the bottleneck in Deep Neural Networks”. In: *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. 2022.

[38] Sahil Verma and Julia Sass Rubin. “Fairness Definitions Explained”. In: *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)* (2018), pp. 1–7.

[39] C. Wah et al. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. CNS-TR-2011-001. California Institute of Technology, 2011.

[40] Kai Yuanqing Xiao et al. “Noise or Signal: The Role of Image Backgrounds in Object Recognition”. In: *International Conference on Learning Representations*. 2021.

[41] Michael Zhang and Christopher Ré. “Contrastive Adapters for Foundation Model Group Robustness”. In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 21682–21697.

[42] Zhilu Zhang and Mert Sabuncu. “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels”. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[43] Bolei Zhou et al. “Places: A 10 Million Image Database for Scene Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1452–1464.

VII. Appendix

A. SkewSize*: computing Cramér's V*

In [3], the authors estimate the effect size via the Cramér's V statistic defined in equation (4). We will precise its content :

1) $N_{y'}$ is the number of samples of class y'
2) $DF_{y'}$ is the number of degrees of freedom
3) $\chi^2_{y'}$ is the test statistic from the Pearson's chi-squared independence test. To compute it, we need to count *among the subset of samples from class* y' the number of predictions $n_{b,y}$ for all of the $M = N_B \cdot N_C$ pairs $(b, \hat{y})$ composed of a bias attribute and a predicted class. Then χ^2 is defined as follows :

$$\chi^2 = \sum_{b,y} \frac{(n_{b,y} - e_{b,y})^2}{e_{b,y}}$$

where $e_{b,y}$ is the expected count of predictions corresponding to the pair (b, y) under the assumption that $B \perp \hat{Y}$, *i.e.*, $e_{b,y} = \frac{n_b \cdot n_y}{N_{y'}}$ where $n_b = \sum_{y \in \dagger} n_{b,y}$ and $n_y = \sum_{b \in \mathcal{B}} n_{b,y}$.

B. *Comprehensive comparison*

Table II provide a more comprehensive overview of the performance of the proposed debiasing methods than table I.

C. *More benchmark datasets*

1) Waterbirds: Introduced in [36], Waterbirds is a image dataset that combines bird photographs from the Caltech-UCSD Birds-200-2011 (CUB) dataset [39] with image backgrounds from the Places dataset [43]. The target attribute is the type of bird $\mathcal{Y}$ ={waterbird, landbird} and the bias attribute is the background $\mathcal{B}$ = {water background, land background}, with waterbirds (resp. landbirds) more frequently appearing against a water (resp. land) background. This dataset provide a benchmark to evaluate the performance of debiasing methods when the type of object of interest correlates with the background.

2) BAR: Introduced in [30], BAR (for Biased Action Recognition) is an action recognition dataset biased towards the setting in which the action takes place. The six action classes are $\mathcal{Y}$ ={climbing, diving, fishing, racing, throwing, vaulting} and the associated places are $\mathcal{B}$ ={rock wall, underwater, water surface, paved track, playing field, sky}. The training set consists of 1941 bias-aligned images with bias labels, while the test set consist of 654 bias-conflicting images without bias labels. Some versions of the dataset introduced a controlled proportion of bias-conflicting samples in the train set, still leaving only bias-conflicting samples in the training set.

BAR constitutes a stark example of the isolated-bias generalization scenario. Indeed, if we consider for instance a bias-conflicting image from the climbing class, the background could be a snowy mountain instead of a rock wall. However, the background snowy mountain is not a background associated with any of the other classes. Hence the biased model might misclassify this sample not because[17] it has the bias information related to another class, but because it does not have the bias information related to the target class. This illustrates the distinction with the other debiasing scenario : cross-bias generalization.

3) Benchmarks based on ImageNet: ImageNet [35] is a large scale collection of more than 14 million natural images with more than 21,000 labels. It is one of the most widely used image dataset to benchmark state of the art techniques and to pre-train models. Debiasing datasets are obtained from ImageNet by retaining specific subsets of images.

9-class ImageNet is a dataset introduced in [40] to test the influence of the background on the classification of the foreground object. As the name implies, it is composed of 9 super-classes of ImageNet ; $\mathcal{Y}$ = {dog, bird, vehicle, reptile, carnivore, insect, instrument, primate, fish } with the bias as the background (unlabeled). 9-class ImageNet contains 45,405 images for training and 4,050 for testing. It comes in 7 versions, among which there is a variant where the foreground stays the same but the background is swapped from a random image from the same class (this yields a *biased-aligned* or *biased* dataset). In another variant, the foreground stays the same but the background is swapped from a random image from another class (this yields a *bias-conflicting* dataset).

ImageNet-A is introduced in [15], as a collection of natural adversarial examples, *i.e.*, , natural images that are misclassified by a large range of models. It contains 7,500 adversarial images from 200 classes. Crucially, the authors focused on clear images (*i.e.*, not cluttered) and on common classes, so that the misclassification is likely to be due to spurious biases, and not to bad quality images. ImageNet-A serves as an unlabeled bias-conflicting test set for classifiers trained on ImageNet.

[17] "because" is an abuse of language, as knowing the causal relationship between the features of an image and the output of a model is difficult. We use it nonetheless for the sake of simplicity, with the goal to convey the intuition.

TABLE II

RESULTS OF THE PRESENTED METHODS ON A SELECTION OF DATASETS. BEST PERFORMING RESULTS ARE MARKED IN BOLD, SECOND BEST ARE UNDERLINED. BEST UNSUPERVISED RESULTS ARE HIGHLIGHTED. WHEN THE SOURCE IS INDICATED NEXT TO A NUMBER, IT MEANS THAT THE VALUE WAS TAKEN FROM THE INDICATED PAPER. WHEN NO SOURCE IS INDICATED, THE VALUE WAS TAKEN FROM THE COLUMN'S PAPER. VALUES SEPARATED BY "/" ARE TAKEN FROM DIFFERENT PAPERS, THEY ARE INDICATED FOR COMPLETENESS BECAUSE THEY DIFFER.

Benchmark	Method	Vanilla ERM	**ϵ-SupInfoNCE** **+FairKL**[7]	**LfF** [30]	**GroupDRO** [36]	**VCBA** [28]	**ReBias** [6]	**DFA** [21]	**EIIL**[10] [10]	**JTT** [24]	**LfF+BiasAdv** [22]
	Type of method	Unsupervised	Supervised	Unsupervised	Supervised	Unsupervised	Prior guided	Unsupervised	Unsupervised	Weakly Supervised	Unsupervised
	Variant										
Biased MNIST (unbiased)	$\rho = 0.99$	88.4 [28] / 90.8 ± 0.3 [7]	**97.86 ± 0.02**	95.1 [28]	-	97.7 ± 0.3	88.1	-	-	-	-
	$\rho = 0.995$	72.4 [28] / 79.5 ± 0.1 [7]	**97.00 ± 0.06**	90.3 [28]	-	95.5 ± 0.8	76	-	-	-	-
	$\rho = 0.997$	40.5 [28] / 62.5 ± 2.9 [7]	**96.19 ± 0.23**	63.7 [28]	-	92.7 ± 1.2	64.2	-	-	-	-
	$\rho = 0.999$	11.2 [28] / 11.8 ± 0.7 [7]	**90.51 ± 1.55**	15.3 [28]	-	58.7 ± 21.8	22.7	-	-	-	-
Colored MNIST (unbiased)	$\rho = 0.95$	82.17 ± 0.74 [21] / 77.63 ± 0.44 [30]	-	85.39 ± 0.94 / 80.57 ± 3.84 [21]	84.50 ± 0.46 [30]	-	**96.96 ± 0.04** [21]	89.66 ± 1.09	-	-	-
	$\rho = 0.98$	65.86 ± 3.59 [21] / 62.29 ± 1.47 [30]	-	80.48 ± 0.45 / 71.03 ± 2.44 [21]	76.30 ± 1.53 [30]	-	**92.91 ± 0.15** [21]	84.79 ± 0.95	-	-	-
	$\rho = 0.99$	52.09 ± 2.88 [21] / 50.34 ± 0.16 [30]	-	74.01 ± 2.21 / 61.89 ± 4.97 [21]	71.33 ± 1.76 [30]	-	**87.4 ± 0.78** [21]	81.73 ± 2.34	-	-	-
	$\rho = 0.995$	35.19 ± 3.49 [21] / 35.34 ± 0.13 [30]	-	63.39 ± 1.97 / 52.50 ± 2.43 [21]	59.67 ± 2.73	-	**70.47 ± 1.84** [21]	65.22 ± 4.41	-	-	-
Waterbirds	Avg group acc	97.3 [36]	-	**97.5** [24]	93.5 ± 0.3	-	-	-	96.9	93.6	-
	Worst group acc	72.6 [24] / 60.3 [10] / 63.7 ± 1.9 [36]	-	75.2 [24]	**91.4 ± 1.1**	-	-	-	78.7	86.0	-
CelebA (target="Blond Hair") (bias="Gender")	unbiased	70.25 ± 0.35 [30] / 79.0 [28]	-	84.24 ± 0.37	85.43 ± 0.53 [30]	**90.2 ± 1.1**	-	-	-	-	-
	bias-conflicting	52.52 ± 0.19 [30] / 59.0 [28]		81.24 ± 1.38	83.40 ± 0.67 [30]	**84.5 ± 2.0**	-	-	-	-	-
CelebA (target="Heavy makeup") (bias="Gender")	unbiased	62.00 ± 0.02 [30]	-	**66.20 ± 1.21**	64.88 ± 0.42 [30]	-	-	-	-	-	-
	bias-conflicting	33.75 ± 0.28 [30]	-	45.48 ± 4.33	**50.24 ± 0.68** [30]	-	-	-	-	-	-
Corrupted CIFAR-10^{0} (unbiased)	$\rho = 0.95$	39.42 ± 0.64 [21] / 37.05 ± 0.37 [22]	50.73 ± 0.90	50.27 ± 1.56 [21] / 50.72 ± 1.31 [22]	-	-	43.43 ± 0.41 [21]	51.13 ± 1.28	-	41.20 ± 0.19 [22]	**57.78 ± 0.33**
	$\rho = 0.98$	30.06 ± 0.71 [21] / 29.66 ± 0.27 [22]	41.45 ± 0.42	39.91 ± 0.30 [21]/40.66 ± 0.70 [22]	-	-	31.66 ± 0.43 [21]	41.78 ± 2.29	-	31.44 ± 0.47 [22]	**48.36 ± 0.59**
	$\rho = 0.99$	25.82 ± 0.33 [21]	**36.53 ± 0.38**	33.07 ± 0.77 [21]	-	-	25.72 ± 0.20 [21]	36.49 ± 1.79	-	-	-
	$\rho = 0.995$	23.08 ± 1.25 [21] / 21.29 ± 0.31 [22]	33.33 ± 0.38	28.57 ± 1.30 [21]/28.81 ± 0.44 [22]	-	-	22.27 ± 0.41 [21]	29.95 ± 0.71	-	23.66 ± 0.78 [22]	**36.78 ± 0.20**
9-class ImageNet	biased set	94.0 ± 0.1 [16]	95.1 ± 0.1	91.2 ± 0.1 [16]	-	**95.5 ± 0.2 / 96.4 ± 0.0**(bagnet)	94.0 ± 0.2 [16]	-	-	-	-
	unbiased set	92.7 ± 0.2 [16]	**94.8 ± 0.3**	89.6 ± 0.3 [16]	-	-	92.7 ± 0.2 [16]	-	-	-	-
ImageNet-A	-	30.5 ± 0.5 [16]	**35.7 ± 0.5**	29.4 ± 0.8 [16]	-	34.2 ± 0.9 / 34.5 ± 3.4 (bagnet)	30.5 ± 0.2 [16]	-	-	-	-
CelebA (target="Blond Hair") (bias="Gender")	Avg group acc	**95.6** [24]	-	86 [24]	92.9 ± 0.2	-	-	-	-	88	-
	Worst group acc	47.2 [24]	-	70.6 [24]	**88.9 ± 2.3**	-	-	-	-	81.1	-
Multicolor MNIST	Unbiased	57.2 [28]	-	52.0 [28]	-	**73.1 ± 0.9**	-	-	69.7 [28]	-	-
	$L^{\|\|} + R^{\|\|}$	**100.0** [28]	-	99.6 [28]	-	**100 ± 0.0**	-	-	**100** [28]	-	-
	$L^{\|\|} + R^{\perp}$	97.1 [28]	-	4.7 [28]	-	90.9 ± 3.5	-	-	**97.2** [28]	-	-
	$L^{\perp} + R^{\|\|}$	27.5 [28]	-	**98.6** [28]	-	77.5 ± 2.8	-	-	70.8 [28]	-	-
	$L^{\perp} + R^{\perp}$	5.2 [28]	-	5.1 [28]	-	**24.1 ± 1.8**	-	-	10.9 [28]	-	-
BAR ($\rho = 1.0$)	avg class acc	51.85 ± 5.92 [30]	-	**62.98 ± 2.76**	-	-	59.74 ± 1.49 [30]	-	-	-	-
	climbing	59.05 ± 17.48 [30]	-	**79.36 ± 4.79**	-	-	77.78 ± 8.32 [30]	-	-	-	-
	diving	16.56 ± 1.58 [30]	-	34.59 ± 2.26	-	-	**51.57 ± 4.54** [30]	-	-	-	-
	fishing	62.69 ± 3.64 [30]	-	**75.39 ± 3.63**	-	-	54.76 ± 2.38 [30]	-	-	-	-
	racing	77.27 ± 2.62 [30]	-	**83.08 ± 1.90**	-	-	80.56 ± 2.19 [30]	-	-	-	-
	throwing	28.62 ± 2.95 [30]	-	**33.72 ± 0.68**	-	-	28.63 ± 2.71 [30]	-	-	-	-
	vaulting	66.92 ± 7.25 [30]	-	**71.75 ± 3.32**	-	-	65.14 ± 7.21 [30]	-	-	-	-
BAR	($\rho = 0.95$)	68.60 ± 2.25 [22]	-	67.48 ± 0.46 [22]	-	-	65.00 [32]	63.5 [32]	-	68.53 ± 3.29 [22]	**72.62 ± 0.11**
	($\rho = 0.99$)	57.65 ± 2.36 [22]	-	57.71 ± 3.12 [22]	-	-	52.1 [32]	52.3 [32]	-	58.17 ± 3.30 [22]	**63.20 ± 2.64**
Colored MNIST (bias-conflicting)	$\rho = 0.95$	77.63 ± 0.44 [30]	-	**85.77 ± 0.66**	83.11 ± 0.41 [30]	-	-	-	-	-	-
	$\rho = 0.98$	62.29 ± 1.47 [30]	-	**80.67 ± 0.56**	74.28 ± 1.90 [30]	-	-	-	-	-	-
	$\rho = 0.99$	50.34 ± 0.16 [30]	-	**74.19 ± 1.94**	69.58 ± 1.66 [30]	-	-	-	-	-	-
	$\rho = 0.995$	35.34 ± 0.13 [30]	-	**63.49 ± 1.94**	57.07 ± 3.60 [30]	-	-	-	-	-	-
Corrupted CIFAR-10^{1} (bias-conflicting)	$\rho = 0.95$	39.42 ± 0.20 [30]	-	**59.62 ± 0.03**	49.00 ± 0.45 [30]	-	-	-	-	-	-
	$\rho = 0.98$	22.65 ± 0.95 [30]	-	**48.69 ± 0.70**	35.10 ± 0.49 [30]	-	-	-	-	-	-
	$\rho = 0.99$	14.24 ± 1.03 [30]	-	**39.55 ± 2.56**	28.04 ± 1.18 [30]	-	-	-	-	-	-
	$\rho = 0.995$	10.50 ± 0.71 [30]	-	**28.61 ± 1.25**	24.40 ± 0.28 [30]	-	-	-	-	-	-
Corrupted CIFAR-10^{2} (bias-conflicting)	$\rho = 0.95$	34.97 ± 1.06 [30]	-	**58.64 ± 1.04**	54.60 ± 0.11 [30]	-	-	-	-	-	-
	$\rho = 0.98$	20.52 ± 0.73 [30]	-	**48.99 ± 1.61**	42.71 ± 1.24 [30]	-	-	-	-	-	-
	$\rho = 0.99$	12.11 ± 0.29 [30]	-	**40.84 ± 2.06**	37.07 ± 1.02 [30]	-	-	-	-	-	-
	$\rho = 0.995$	10.01 ± 0.01 [30]	-	**32.03 ± 2.51**	30.92 ± 0.86 [30]	-	-	-	-	-	-
Corrupted CIFAR-10^{1} (unbiased)	$\rho = 0.95$	45.24 ± 0.22 [30]	-	**85.39 ± 0.94**	53.15 ± 0.53 [30]	-	-	-	-	-	-
	$\rho = 0.98$	30.21 ± 0.82 [30]	-	**80.48 ± 0.45**	40.19 ± 0.23 [30]	-	-	-	-	-	-
	$\rho = 0.99$	22.72 ± 0.87 [30]	-	**74.01 ± 2.21**	32.11 ± 0.83 [30]	-	-	-	-	-	-
	$\rho = 0.995$	17.93 ± 0.66 [30]	-	**63.39 ± 1.97**	29.26 ± 0.11 [30]	-	-	-	-	-	-
Corrupted CIFAR-10^{2} (unbiased)	$\rho = 0.95$	41.27 ± 0.98 [30]	-	**85.39 ± 0.94**	57.92 ± 0.31 [30]	-	-	-	-	-	-
	$\rho = 0.98$	28.29 ± 0.62 [30]	-	**80.48 ± 0.45**	46.12 ± 1.11 [30]	-	-	-	-	-	-
	$\rho = 0.99$	20.71 ± 0.29 [30]	-	**74.01 ± 2.21**	39.57 ± 1.04 [30]	-	-	-	-	-	-
	$\rho = 0.995$	17.37 ± 0.31 [30]	-	**63.39 ± 1.97**	34.25 ± 0.74 [30]	-	-	-	-	-	-