Fairness Through Robustness: Investigating Robustness Disparity in Deep Learning

Vedant Nanda*, Samuel Dooley*, Sahil Singla, Soheil Feizi, and John P Dickerson

University of Maryland, College Park

Abstract

Deep neural networks are being increasingly used in real world applications (e.g. surveillance, face recognition). This has resulted in concerns about the fairness of decisions made by these models. Various notions and measures of fairness have been proposed to ensure that a decision-making system does not disproportionately harm (or benefit) particular subgroups of population. In this paper, we argue that traditional notions of fairness that are only based on models' outputs are not sufficient when decision-making systems such as deep networks are vulnerable to adversarial attacks. We argue that in some cases, it may be easier for an attacker to target a particular subgroup, resulting in a form of *robustness bias*. We propose a new notion of *adversarial fairness* that requires all subgroups to be equally robust to adversarial perturbations. We show that state-of-the-art neural networks can exhibit robustness bias on real world datasets such as CIFAR10, CIFAR100, Adience, and UTKFace. We then formulate a measure of our proposed fairness notion and use it as a regularization term to decrease the robustness bias in the traditional empirical risk minimization objective. Through empirical evidence, we show that training with our proposed regularization term can partially mitigate adversarial unfairness while maintaining reasonable classification accuracy.

1 Introduction

Automated decision-making systems that are driven by data are being used in a variety of different real-world applications. In many cases, these systems make decisions on data points that represent humans (*e.g.*, targeted ads [39], personalized recommendations [3, 37], hiring [34, 35], credit scoring [24], or recidivism prediction [8]). In such scenarios, there is often concern regarding the fairness of outcomes of the systems [2]. This has resulted in a growing body of work from the nascent Fairness, Accountability, Transparency, and Ethics (FATE) community that—drawing on prior legal and philosophical doctrine—aims to define, measure, and (attempt to) mitigate manifestations of unfairness in automated systems [4, 8, 13, 27].

Most of the initial work on fairness in machine learning considered notions that were one-shot and considered the model and data distribution to be static [2, 8, 11, 43–45]. Recently, there has been more work exploring notions of fairness that are dynamic and consider the possibility that the world (*i.e.*, the model as well as data points) might change over time [18, 20, 21, 28]. Our proposed notion of robustness bias has subtle difference from existing one-shot and dynamic notions of fairness in that it requires each partition of the population be equally robust to imperceptible changes in the input (*e.g.*, noise, adversarial perturbations, etc).

We propose a simple and intuitive notion of *robustness bias* which requires subgroups of populations to be equally "robust." Robustness can be defined in multiple different ways [14, 30, 41]. We take a general definition which assigns points that are farther away from the decision boundary higher robustness. Our key contributions are as follows:

*Denotes Equal Contribution – Emails: {vedant,sdooley1,ssingla,sfeizi,john}@cs.umd.edu

- We define a simple, intuitive notion of ***robustness bias*** that requires all partitions of the dataset to be equally robust. We argue that such a notion is especially important when the decision-making system is a deep neural network (DNN) since these have been shown to be susceptible to various attacks [7, 29]. Importantly, our notion depends not only on the outcomes of the system, but also on the distribution of distances of data-points from the decision boundary, which in turn is a characteristic of *both* the data distribution and the learning process.
- We propose different methods to ***measure this form of bias***. Measuring the exact distance of a point from the decision boundary is a challenging task for deep neural networks which have a highly non-convex decision boundary. This makes the measurement of robustness bias a non-trivial task. In this paper we use existing ways of approximating the distance of each point (*e.g.*, adversarial attacks).
- We do an in-depth analysis of a special case of *robustness bias* which requires all partitions (or subgroups) of the dataset (often described by some sensitive feature like race, gender *etc.*) to be ***equally robust to adversarial perturbations***. Through ***extensive empirical evaluation*** we show that adversarial unfairness can exist for many state-of-the art models that are trained on common classification datasets. We raise several questions regarding the fairness of such models when it is be easier to attack a certain population through adversarial perturbations.
- Finally, we propose a novel ***regularization term*** which captures our proposed notion. We show that this can be used as a regularizer to reduce robustness bias.

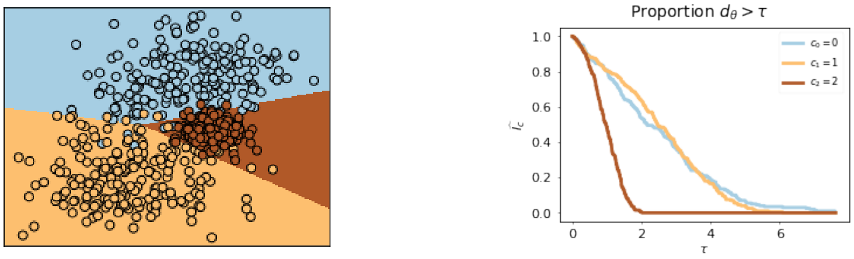
1.1 Related Work

Fairness in ML. Models that learn from historic data have been shown to exhibit unfairness, *i.e.*, they disproportionately benefit or harm certain subgroups (often a sub-population that shares a common sensitive attribute such as race, gender *etc.*) of the population [2, 8, 24]. This has resulted in a lot of work on quantifying, measuring and to some extent also mitigating unfairness [1, 10, 11, 15, 17, 32, 42–45]. Most of these works consider notions of fairness that are one-shot—that is, they do not consider how these systems would behave over time as the world (*i.e.*, the model and data distribution) evolves. Recently more works have taken into account the dynamic nature of these decision-making systems and consider fairness definitions and learning algorithms that fare well across multiple time steps [18, 20, 21, 28]. We take inspiration from both the one-shot and dynamic notions, but take a slightly different approach by requiring all subgroups of the population to be equally robust to minute changes in their features. These changes could either be random or carefully crafted adversarial noise. This is closely related to Heidari et al. [21]'s effort-based notion of fairness; however, their notion has a very specific use case of societal scale models whereas our approach is more general and applicable to all kinds of models. Our work is also closely related to and inspired by Zafar et al.'s use of a regularized loss function which captures fairness notions and reduces disparity in outcomes [44].

Adversarial Attacks. Deep Neural Networks (DNNs) have been shown to be susceptible to carefully crafted adversarial perturbations which—imperceptible to a human—result in a misclassification by the model [14, 30, 41]. In the context of our paper, we use adversarial attacks to approximate the distance of a data point to the decision boundary. For this we use state-of-the-art white-box attacks proposed by Moosavi-Dezfooli et al. [29] and Carlini and Wagner [7]. Due to the many works on adversarial attacks, there have been many recent works on provable robustness to such attacks. The high-level goal of these works is to estimate a (tight) lower bound on the distance of a point from the decision boundary [9, 33, 38]. We leverage these methods to estimate distances from the decision boundary which helps assess robustness bias (defined formally in Section 3).

2 Heterogeneous Susceptibility to Adversarial Attacks

In a classification setting, a learner is given some data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ consisting of inputs $x_i \in \mathbb{R}^d$ and outputs $y_i \in \mathcal{C}$ which are labels in some set of classes $\mathcal{C} = \{c_1, \ldots, c_k\}$. These classes form a partition on the dataset such that $\mathcal{D} = \bigsqcup_{c \in \mathcal{C}} \{(x_i, y_i) \mid y_i = c_j\}$. The goal of learning in decision boundary-based optimization is to draw delineations between points in feature space which sort the data into groups according to their class label. The learning generally tries to maximize the classification accuracy of the decision boundary choice. A learner chooses some loss function $\mathcal{L}$ to



(a) Three-class classification problem for randomly generated data.

(b) Proportion of a class which is greater than τ away from a decision boundary.

Figure 1: An example of multinomial logistic regression.

minimize on a training dataset, parameterized by parameters θ, while maximizing the classification accuracy on a test dataset.

Of course there are other aspects to classification problems that have recently become more salient in the machine learning community. Considerations about the fairness of classification decisions, for example, are one such way in which additional constraints are brought into a learner's optimization strategy. In these settings, the data $\mathcal{D} = \{(x_i, y_i, s_i)\}_{i=1}^N$ is imbued with some metadata which have a sensitive attribute $\mathcal{S} = \{s_1, \dots, s_t\}$ associated with each point. Like the classes above, these sensitive attributes form a partition on the data such that $\mathcal{D} = \bigsqcup_{s \in \mathcal{S}} \{(x_i, y_i, s_i) \mid s_i = s\}$. Without loss of generality, we assume a single sensitive attribute. Generally speaking, learning with fairness in mind considers the output of a classifier based off of the partition of data by the sensitive attribute, where some objective behavior, like minimizing disparate impact or treatment [44], is integrated into the loss function or learning procedure to find the optimal parameters θ.

There is not a one-to-one correspondence between decision boundaries and classifier performance. For any given performance level on a test dataset, there are infinitely many decision boundaries which produce the same performance, see Figure 2. This raises the question: *if we consider all decision boundaries or model parameters which achieve a certain performance, how do we choose among them? What are the properties of a desirable, high-performing decision boundary?* As the community has discovered, one *undesirable* characteristic of a decision boundary is its proximity to data which might be susceptible to adversarial attack [14, 30, 41]. This provides intuition that we should prefer boundaries that are as far away as possible from example data [5, 40].

Let us look at how this plays out in a simple example. In multinomial logistic regression, the decision boundaries are well understood and can be written in closed form. This makes it easy for us to compute how close each point is to a decision boundary. Consider for example a dataset and learned classifier as in Figure 1a. For this dataset, we observe that the brown class, as a whole, is closer to a decision boundary than the yellow or blue classes. We can quantify this by plotting the proportion of data that are greater than a distance τ away from a decision boundary, and then varying τ. Let $d_\theta(x)$ be the minimal distance between a point x and a decision boundary corresponding to parameters θ. For a given partition $\mathcal{P}$ of a dataset, $\mathcal{D}$, such that $\mathcal{D} = \bigsqcup_{P \in \mathcal{P}} P$, we define the function:

$$\widehat{I_P}(\tau) = \frac{|\{(x, y) \in P \mid d_\theta(x) > \tau, y = \hat{y}\}|}{|P|}$$

If each element of the partition is uniquely defined by an element, say a class label, c, or a sensitive attribute label, s, we equivalently will write $\widehat{I_c}(\tau)$ or $\widehat{I_s}(\tau)$ respectively. We plot this over a range of τ in Figure 1b for the toy classification problem in Figure 1a. Observe that the function for the brown class decreases significantly faster than the other two classes, quantifying how much closer the brown class is to the decision boundary.

From a strictly classification accuracy point of view, the brown class being significantly closer to the decision boundary is not of concern; all three classes achieve similar classification accuracy. However, when we move away from this toy problem and into neural networks on real data, this difference between the classes could become a potential vulnerability to exploit, particularly when we consider adversarial examples.

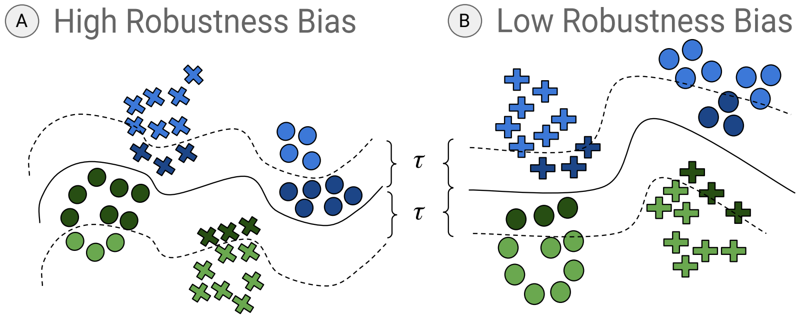


Figure 2: A toy example showing robustness bias. A.) the classifier (solid line) has 100% accuracy for blue and green points. However for a budget τ (dotted lines), 70% of points belonging to the "round" subclass (showed by dark blue and dark green) will get attacked while only 30% of points in the "cross" subclass will be attacked. This shows a clear bias against the "round" subclass which is less robust in this case. B.) shows a different classifier for the same data points also with 100% accuracy. However, in this case, with the same budget τ, 30% of both "round" and "cross" subclass will be attacked, thus being less biased.

3 Robustness Bias

Our goal is to understand how susceptible different classes are to adversarial attacks. Ideally, no one class would be more susceptible than any other, but this may not be possible. We have observed that for the same dataset, there may be some classifiers which have differences between the distance of that partition to a decision boundary; and some which do not. There may also be one partition $\mathcal{P}$ which exhibits this discrepancy, and another partition $\mathcal{P}'$ which does not. Therefore, we make the following statement about robustness bias:

Definition 1. *A dataset $\mathcal{D}$ with a partition $\mathcal{P}$ and a classifier parameterized by θ exhibits* **robustness bias** *if there exists an element of $\mathcal{P}$ for which the elements of that partition are either significantly closer to (or significantly farther from) a decision boundary than elements not in P.*

We might say that a dataset, partition, and classifier do not exhibit robustness bias if for all $P, P' \in \mathcal{P}$ and all $\tau > 0$

$$\mathbb{P}_{(x,y)\in\mathcal{D}}\{d_\theta(x) > \tau \mid x \in P, y = \hat{y}\} \approx \mathbb{P}_{(x,y)\in\mathcal{D}}\{d_\theta(x) > \tau \mid x \in P', y = \hat{y}\}.$$

Intuitively, this definition requires that for a given perturbation budget τ and a given partition P, one should not have any incentive to perturb data points from P over points that do not belong to P. Even when examining this criteria, we can see that this might be particularly hard to satisfy. Thus, we want to quantify the disparate susceptibility of each element of a partition to adversarial attack, i.e., how much farther or closer it is to a decision boundary when compared to all other points. We can do this with the following function for a dataset $\mathcal{D}$ with partition element $P \in \mathcal{P}$ and classifier parameterized by θ:

$$RB(P, \tau) = |\,\mathbb{P}_{x\in\mathcal{D}}\{d_\theta(x) > \tau \mid x \in P, y = \hat{y}\} - \mathbb{P}_{x\in\mathcal{D}}\{d_\theta(x) > \tau \mid x \notin P, y = \hat{y}\}\,|$$

Observe that $RB(P, \tau)$ is a large value if and only if the elements of P are much more (or less) adversarially robust than elements not in P. We can then quantify this for each element $P \in \mathcal{P}$—but a more pernicious variable to handle is τ. We propose to look at the area under the curve $\widehat{I_P}$ for all τ:

$$\sigma(P) = \frac{AUC(\widehat{I_P}) - AUC(\sum_{P' \neq P} \widehat{I_{P'}})}{AUC(\sum_{P' \neq P} \widehat{I_{P'}})}$$

Note that these notions take into account the distances of data points from the decision boundary and hence are orthogonal and complementary to other traditional notions of bias or fairness (*e.g.*, disparate impact/disparate mistreatment [44], etc). This means that having lower robustness bias does not necessarily come at the cost of fairness as measured by these notions. Consider the motivating example shown in Figure 2: the decision boundary on the right has lower robustness bias but preserves all other common notions (*e.g.* [11, 17, 43]) as both classifiers maintain 100% accuracy.

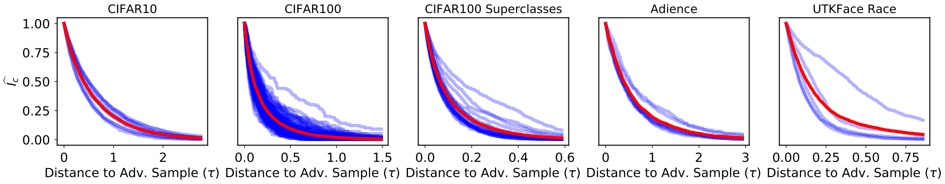


Figure 3: For each dataset, we plot $\widehat{I_c}(\tau)$ for each class c in each dataset. Each blue line represents one class. The red line represents the mean of the blue lines, i.e., $\sum_{c \in C} \widehat{I_c}(\tau)$ for each τ.

3.1 Using Upper Bounds to Estimate $d_\theta(x)$, the Distance to the Decision Boundary

The above requires a way to measure the distance between a point and the closest decision boundary. For most classifiers in use today, a direct computation of $d_\theta(x)$ is not feasible. However, there are efficient ways to approximate $d_\theta(x)$ for a given neural network. One such way is to use an adversarial example, yielding an upper bound on the distance; another is to use techniques from the robustness certificates literature [9, 33, 38] to yield a lower bound on the distance. From a fairness standpoint, upper bounds are more appropriate since perturbing an input by τ where τ is the upper bound for that input will provably result in a different classification.

For a given input and model, one can compute an adversarial example by performing an optimization which alters the input image slightly so as to place the altered image into a different category than the original. Assume for a given datapoint x, we are able to compute an adversarial image $\tilde{x}$, then the distance between these two images provides an upper bound on minimal distance to a decision boundary, i.e, $\|x - \tilde{x}\| \geq d_\theta(x)$.

We evaluate two adversarial attacks: DeepFool [29] and CarliniWagner's L2 attack [7]. We extend $\widehat{I_P}$ for DeepFool as $\widehat{I_P^{DF}} = \frac{|\{(x,y) \in P | \tau < \|x - \tilde{x}\|, y = \hat{y}\}|}{|P|}$. We use similar notation to define $\sigma^{DF}(P)$, $\widehat{I_P^{CW}}$, and $\sigma^{CW}(P)$. While these methods are guaranteed to yield upper bounds on $d_\theta(x)$, they need not yield similar behavior to $\widehat{I_P}$ or $\sigma(P)$. We perform an evaluation of this in Section 4.3.

4 Experimental Evidence Robustness Bias Exists in the Wild

We hypothesize that there exist datasets and model architectures which exhibit adversarial robustness bias. To investigate this claim, we examine several image-based classification datasets and common model architectures.

4.1 Datasets and Model Architectures

We perform these tests of the datasets **CIFAR-10** [25], **CIFAR-100** [25] (using both 100 classes and 20 super classes), **Adience** [12], and **UTKFace** [46]. The first two are widely accepted benchmarks in image classification, while the latter two provide significant metadata about each image, permitting various partitions of the data by final classes and sensitive attributes.

Our experiments were performed using PyTorch's torchvision module [31]. We first explore a simple **Multinomial Logistic Regression** model which could be fully analyzed with direct computation of the distance to the nearest decision boundary. For convolutional neural networks, we focus on **Alexnet** [26], **VGG** [36], **ResNet** [19], **DenseNet** [22], and **Squeezenet** [23] which are all available through torchvision. We use these models since these are widely used for a variety of tasks. We achieve performance that is comparable to state of the art performance on these datasets for these models [1].

4.2 Exact Computation in a Simple Model: Multinomial Logistic Regression

We begin our analysis by studying the behavior of multinomial logistic regression. Admittedly, this is a simple model compared to modern deep-learning-based approaches; however, it enables is to

[1] See Appendix A Table 1 for model performances, additional quantitative results and supporting figures.

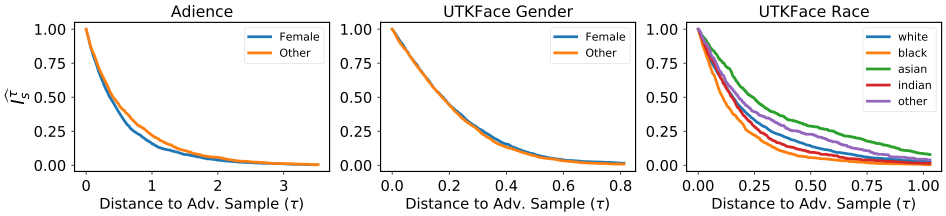


Figure 4: For each dataset, we plot $\widehat{I_s^\tau}$ for each sensitive attribute s in each dataset. Additional figures for other models can be found in Appendix A.

explicitly compute the exact distance to a decision boundary, $d_\theta(x)$. We fit a regression to each of our vision datasets to their native classes and plot $\widehat{I_c}(\tau)$ for each dataset. Figure 3 shows the distributions of $\widehat{I_c}(\tau)$, from which we observe three main phenomena: (1) the general shape of the curves are similar for each dataset, (2) there are classes which are significant outliers from the other classes, and (3) the range of support of the τ for each dataset varies significantly. We discuss each of these individually.

First, we note that the shape of the curves for each dataset is qualitatively similar. Since the form of the decision boundaries in multinomial logistic regression are linear delineations in the input space, it is fair to assume that this similarity in shape in Figure 3 can be attributed to the nature of the classifier.

Second, there are classes c which indicate disparate treatment under $\widehat{I_c}(\tau)$. The treatment disparities are most notable in UTKFace, the superclass version CIFAR100, and regular CIFAR100. This suggests that, when considering the dataset as a whole, these outlier classes are less suceptible to adversarial attack than other classes. Further, in UTKFace, there are some classes that are considerably more susceptible to adversarial attack because a larger proportion of that class is closer to the decision boundaries.

We also observe that the median distance to decision boundary can vary based on the dataset. The median distance to a decision boundary for each dataset is: 0.40 for CIFAR10; 0.10 for CIFAR100; 0.06 for the superclass version of CIFAR100; 0.38 for Adience; and 0.12 for UTKFace. This is no surprise as $d_\theta(x)$ depends both on the location of the data points (which are fixed and immovable in a learning environment) and the choice of architectures/parameters.

Finally, we consider another partition of the datasets. Above, we consider the partition of the dataset which occurs by the class labels. With the Adience and UTKFace datasets, we have an additional partition by sensitive attributes. Adience admits partitions based off of gender; UTKFace admits partition by gender and ethnicity. We note that Adience and UTKFace use categorical labels for these multidimensional and socially complex concepts. We know this to be reductive and serves to minimize the contextualization within which race and gender derive their meaning [6, 16]. Further, we acknowledge the systems and notions that were used to reify such data partitions and the subsequent implications and conclusions draw therefrom. We use these socially and systemically-laden partitions to demonstrate that the functions we define, $\widehat{I_P}$ and σ depend upon how the data are divided for analysis. To that end, the function $\widehat{I_P}$ is visualized in Figure 4. We observe that the Adience dataset, which exhibited some adversarial robustness bias in the partition on $\mathcal{C}$ only exhibits minor adversarial robustness bias in the partition on $\mathcal{S}$ for the attribute 'Female'. On the other hand, UTKFace which had signifiant adversarial robustness bias does exhibit the phenomenon for the sensitive attribute 'Black' but not for the sensitive attribute 'Female'.

This emphasizes that adversarial robustness bias is dependant upon the dataset and the partition. We will demonstrate later that it is also dependant on the choice of classifier. First, we talk about ways to approximate $d_\theta(x)$ for more complicated models.

4.3 Evaluation of $\widehat{I_P^{DF}}$ and $\widehat{I_P^{CW}}$

To compare the estimate of $d_\theta(x)$ by DeepFool and CarliniWagner, we first look at the signedness of $\sigma(P)$, $\sigma^{DF}(P)$, and $\sigma^{CW}(P)$. Considering all 151 possible partitions for all five datasets, both CarliniWagner and DeepFool agree with the signedness of the direct computation 125 times,

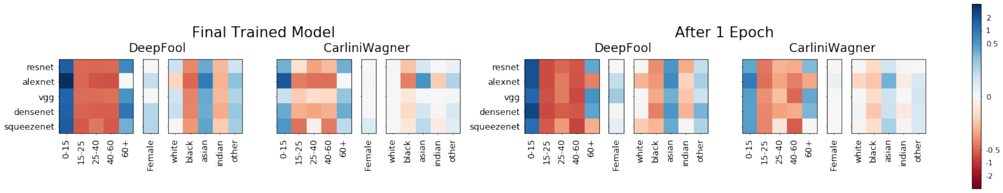


Figure 5: Depiction of σ_P^{DF} and σ_P^{CW} for the UTKFace dataset with partitions corresponding to the (1) class labels $\mathcal{C}$ and the, (2) gender, and (3) race/ethnicity. These values are reported for all five convolutional models both at the beginning of their training (after one epoch) and at the end. We observe that, largely, the signedness of the functions are consistent between the five models and also across the training cycle.

i.e., $\mathbb{1}_P \left[\text{sign}(\sigma(P)) = \text{sign}(\sigma^{DF}(P))\right] = 125 = \mathbb{1}_P \left[\text{sign}(\sigma(P)) = \text{sign}(\sigma^{CW}(P))\right]$. Further, the mean difference between $\sigma(P)$ and $\sigma^{CW}(P)$ or $\sigma^{DF}(P)$, i.e., $(\sigma(P) - \sigma^{DF}(P))$, is 0.17 for DeepFool and 0.19 for CarliniWagner with variances of 0.07 and 0.06 respectively.

There is 83% agreement between the direct computation and the DeepFool and CarliniWagner estimates of $\widehat{I_P}$. This behavior provides evidence that adversarial attacks provide meaningful upper bounds on $d_\theta(x)$ in terms of the behavior of identifying instances of adversarial robustness bias.

4.4 Approximate Computation in Deep Models: CNNs

We now evaluate five commonly-used convolutional networks: Alexnet, VGG, ResNet, DenseNet, and Squeezenet. We trained these networks using PyTorch with standard stochastic gradient descent. We achieve comparable performance to documented state of the art for these models on these datasets. A full table of performance on the test data are described in Table 1. After training each model on each dataset, we generated adversarial examples using both methods and computed σ_P for each possible partition of the dataset. An example of the results for the UTKFace dataset can be see in Figure 5.[2].

With evidence from section 4.3 that DeepFool and CarliniWagner can approximate the robustness bias behavior of direct computations of d_θ, we first ask if there are any major differences between the two methods. *If DeepFool exhibits adversarial robustness bias for a dataset and a model and a class, does CarliniWagner exhibit the same? and vice versa?* Since there are 5 different convolutional models, we have $151 \cdot 5 = 755$ different comparisons to make. Again, we first look at the signedness of $\sigma^{DF}(P)$ and $\sigma^{CW}(P)$ and we see that $\mathbb{1}_P \left[\text{sign}(\sigma^{DF}(P)) = \text{sign}(\sigma^{CW}(P))\right] = 708$. This means there is 94% agreement between DeepFool and CarliniWagner about the direction of the adversarial robustness bias.

To investigate if this behavior is exhibited earlier in the training cycle than at the final, fully-trained model, we compute $\sigma^{CW}(P)$ and $\sigma^{DF}(P)$ for the various models and datasets for trained models after 1 epoch and the middle epoch. For the first epoch, 637 of the 755 partitions were internally consistent, i.e., the signedness of σ was the same in the first and last epoch, and 621 were internally consistent. We see that at the middle epoch, 671 of the 755 partitions were internally consistent for DeepFool and 665 were internally consistent for CarliniWagner. Unsurprisingly, this implies that as the training progresses, so does the behavior of the adversarial robustness bias. However, it is surprising that much more than 80% of the final behavior is determined after the first epoch, and there is a slight increase in agreement by the middle epoch.

We note that, of course, adversarial robustness bias is not necessarily an intrinsic value of a dataset; it may be exhibited by some models and not by others. However, in our studies, we see that the UTKFace dataset partition on Race/Ethnicity does appear to be significantly prone to adversarial attacks given its comparatively low $\sigma^{DF}(P)$ and $\sigma^{CW}(P)$ values across all models.

5 Partially Combatting Robustness Bias through Regularization

Motivated by evidence (see Section 4) that robustness bias exists in a diverse set of real-world models and datasets, we will now show that the expression of robustness bias can be included in an

[2] Our full slate of approximation results are available in Appendix A

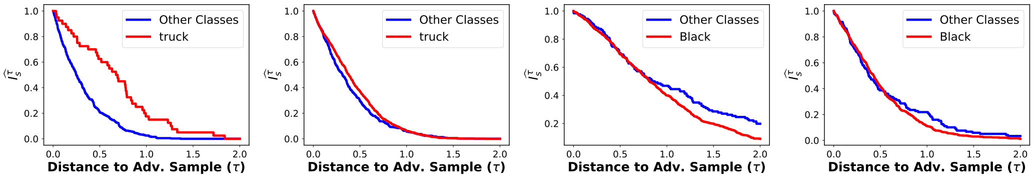
optimization. We do so in a natural way: by formulating a regularization term that captures robustness bias.

Recall the traditional Empirical Risk Minimization objective, ERM $:= l_{cls}(f_\theta(X), Y)$ where l_{cls} is cross entropy loss. Now we wish to model our measure of fairness (see Section 3) and minimize for it alongside ERM. We first write the empiric estimate of $RB(P,\tau)$ as $\tilde{RB}(P,\tau)$; a full derivation can be found in Appendix B.1. Formally,

$$\tilde{RB}(P,\tau) = \left| \frac{1}{\sum\limits_{x \notin P} \mathbb{1}\{y = \hat{y}\}} \sum_{\substack{x \notin P \\ y = \hat{y}}} \mathbb{1}\{d_\theta(x) > \tau\} - \frac{1}{\sum\limits_{x \in P} \mathbb{1}\{y = \hat{y}\}} \sum_{\substack{x \in P \\ y = \hat{y}}} \mathbb{1}\{d_\theta(x) > \tau\} \right|$$

To use this as a regularizer during training, we must compute the closed form expression of d_θ. For this, we take inspiration from the way adversarial inputs are created using DeepFool [29]. Just like in DeepFool, we also approximate distance from f_θ considering f_θ to be linear (even though it may be a highly non-linear DNN). There, we incorporate into $\tilde{RB}(P,\tau)$ the approximation $d_\theta(x) = \left| \frac{f_\theta(x)}{||\nabla_x f_\theta(x)||} \right|$. Finally, we minimize for the new objective function, AdvERM $:= l_{cls}(f_\theta(X), Y) + \alpha RB(P,\tau)$.

5.1 Experimental Results using Regularized Models



(a) CIFAR10 (no reg.) **(b)** $\alpha = 0.1, \tau = 5.0.$ **(c)** UTKFace (no reg.) **(d)** $\alpha = 1.0, \tau = 5.0.$

Figure 6: In the unregularized model, "truck" in CIFAR10 tends to be more robust than other classes (6a); however, using AdvERM reduces that disparity (6b). We see similar behavior for UTKFace (6c & 6d).

Using our regularized objective, AdvERM, we re-train the model (details in Appendix B) considering α and τ as hyperparameters, in addition to the traditional hyperparameters such as learning rate, momentum etc (see Appendix B for reproducibility details). Here, we evaluate the regularized model, after training for a number of vaues of α and τ, for CIFAR10 considering each class as the partition $\mathcal{P}$, and UTKFace with the sensitive attribute race as a partition $\mathcal{P}$. Figure 6 shows example results for the regularized models for the class "truck" in CIFAR10 and for the class "black" in UTKFace. Results for additional classes, datasets, and models can be found in Appendix B.2.

Across the two datasets, we see that for an appropriate α and τ, we are able to reduce the robustness disparity—*i.e.*, difference between blue and red curves—that existed in the original model. For these two datasets, this does not come at any cost of accuracy. We observe test set accuracy of 86.97% on CIFAR10 without regularization and 87.82% with regularization. Similarly for UTKFace we see accuracies of 66.25% and 65.52% for no regularization and with regularization respectively. Results for other models and hyperparameters can be found in Appendix B.2. We interpret our experiments as an indication that an optimization-based approach can play a part in a larger robustness bias-mitigation strategy, rather than serving as panacea.

Broader Impact

We propose a unique definition of fairness which requires all partitions of a population to be equally robust to minute (often adversarial) perturbations, and give experimental evidence that this phenomenon can exist in some commonly-used models trained on real-world datasets. Using these observations, we argue that this can result in a potentially unfair circumstance where, in the presence of an adversary, a certain partition might be more susceptible (*i.e.*, less secure). Thus, we call for extra caution while deploying deep neural nets in the real world since this form of unfairness might go unchecked when auditing for notions that are based on just the model outputs and ground truth labels. We then show that this form of bias can be mitigated to some extent by using a regularizer that minimizes our proposed measure of robustness bias. However, we do not claim to "solve" unfairness; rather, we view analytical approaches to bias detection and optimization-based approaches to bias mitigation as potential pieces in a much larger, multidisciplinary approach to addressing these issues in fielded systems.

Indeed, we view our work as largely observational—we *observe* that, on many commonly-used models trained on many commonly-used datasets, a particular notion of bias, *robustness bias*, exists. We show that some partitions of data are more susceptible to two state-of-the-art and commonly-used adversarial attacks. This knowledge could be used for *attack* or to design *defenses*, both of which could have potential positive or negative societal impacts depending on the parties involved and the reasons for attacking and/or defending. We have also *defined* a notion of bias as well as a corresponding notion of fairness, and by doing that we admittedly toe a morally-laden line. Still, while we do use "fairness" as both a higher-level motivation and a lower-level quantitative tool, we have tried to remain ethically neutral in our presentation and have eschewed making normative judgements to the best of our ability.

Acknowledgments

Dickerson, Dooley, and Nanda were supported in part by NSF CAREER Award IIS-1846237, DARPA GARD #HR00112020007, DARPA SI3-CMD #S4761, DoD WHS Award #HQ003420F0035, and a Google Faculty Research Award. Feizi and Singla were supported in part by the NSF CAREER award 1942230, Simons Fellowship on Deep Learning Foundations, AWS Machine Learning Research Award, and award HR001119S0026-GARD-FP-052. The authors would like to thank Juan Luque and Aviva Prins for fruitful discussions in earlier stages of the project.

References

[1] Tameem Adel, Isabel Valera, Zoubin Ghahramani, and Adrian Weller. One-network adversarial fairness. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 2412–2420, 2019.

[2] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *California Law Review*, 104: 671, 2016.

[3] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *ACM Conference on Research and Development in Information Retrieval (SIGIR)*, page 405–414, 2018.

[4] Reuben Binns. Fairness in machine learning: Lessons from political philosophy. *Proceedings of Machine Learning Research*, 81:1–11, 2017.

[5] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Conference on Learning Theory (COLT)*, page 144–152, 1992.

[6] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 77–91, 2018.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (S&P)*, pages 39–57, 2017.

[8] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017.

[9] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019.

[10] Cynthia Dwork and Christina Ilvento. Fairness under composition. In *Innovations in Theoretical Computer Science Conference (ITCS)*, 2018.

[11] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference (ITCS)*, 2012.

[12] Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, 2014.

[13] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 259–268, 2015.

[14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[15] Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P. Gummadi, and Adrian Weller. Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[16] Alex Hanna, Emily Denton, Andrew Smart, and Jamila Smith-Loud. Towards a critical race methodology in algorithmic fairness. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 501–512, 2020.

[17] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

[18] Tatsunori B. Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning (ICML)*, 2018.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[20] Hoda Heidari and Andreas Krause. Preventing disparate treatment in sequential decision making. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[21] Hoda Heidari, Vedant Nanda, and Krishna P. Gummadi. On the long-term impact of algorithmic decision policies: Effort unfairness and feature segregation through social learning. In *International Conference on Machine Learning (ICML)*, 2019.

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5mb model size. *CoRR*, abs/1602.07360, 2016.

[24] Amir E. Khandani, Adlar J. Kim, and Andrew W. Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787, 2010.

[25] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

[26] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *CoRR*, abs/1404.5997, 2014.

[27] Derek Leben. Normative principles for evaluating fairness in machine learning. In *Conference on Artificial Intelligence, Ethics, and Society (AIES)*, pages 86–92, 2020.

[28] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *International Conference on Machine Learning (ICML)*, 2018.

[29] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

[30] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037. 2019.

[32] Debjani Saha, Candice Schumann, Duncan C. McElfresh, John P. Dickerson, Michelle L Mazurek, and Michael Carl Tschantz. Measuring non-expert comprehension of machine learning fairness metrics. In *International Conference on Machine Learning (ICML)*, 2020.

[33] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 11292–11303. 2019.

[34] Candice Schumann, Samsara N. Counts, Jeffrey S. Foster, and John P. Dickerson. The diverse cohort selection problem. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, page 601–609, 2019.

[35] Candice Schumann, Jeffrey S. Foster, Nicholas Mattei, and John P. Dickerson. We need fairness and explainability in algorithmic hiring. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, page 1716–1720, 2020.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[37] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2018.

[38] Sahil Singla and Soheil Feizi. Second-order provable defenses against adversarial attacks. In *International Conference on Machine Learning (ICML)*, 2020.

[39] Till Speicher, Muhammad Ali, Giridhari Venkatadri, Filipe Nunes Ribeiro, George Arvanitakis, Fabrício Benevenuto, Krishna P. Gummadi, Patrick Loiseau, and Alan Mislove. Potential for discrimination in online targeted advertising. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2018.

[40] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, June 1999.

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[42] Christina Wadsworth, Francesca Vera, and Chris Piech. Achieving fairness through adversarial learning: an application to recidivism prediction. *CoRR*, abs/1807.00199, 2018.

[43] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, Krishna P. Gummadi, and Adrian Weller. From parity to preference-based notions of fairness in classification. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[44] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 20(75):1–42, 2019.

[45] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning (ICML)*, pages 325–333, 2013.

[46] Song Yang Zhang, Zhifei and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

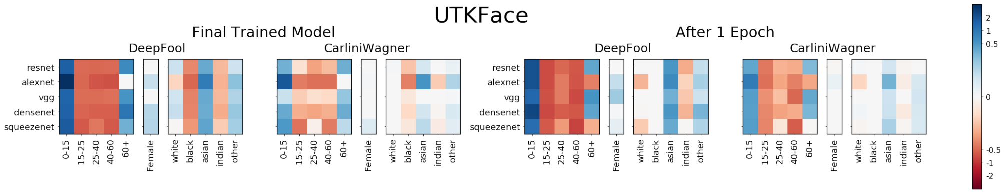


Figure 7: Depiction of σ_P^{DF} and σ_P^{CW} for the UTKFace dataset with partitions corresponding to the (1) class labels $\mathcal{C}$ and the, (2) gender, and (3) race/ethnicity. These values are reported for all five convolutional models both at the beginning of their training (after one epoch) and at the end. We observe that, largely, the signedness of the functions are consistent between the five models and also across the training cycle.

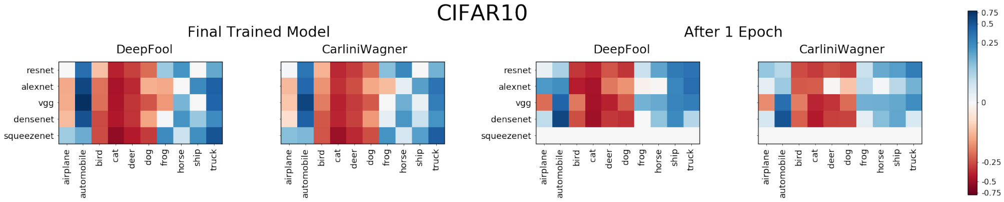


Figure 8: Depiction of σ_P^{DF} and σ_P^{CW} for the CIFAR10 dataset with partitions corresponding to the class labels $\mathcal{C}$. These values are reported for all five convolutional models both at the beginning of their training (after one epoch) and at the end. We observe that, largely, the signedness of the functions are consistent between the five models and also across the training cycle.

A Model Performance

We choose popular models that are widely used by the community for classification tasks on CIFAR-10, CIFAR-100, Adience, and UTKFace. Accuracy of these models can be found in Table 1. All the models are pre-trained on Imagenet and then fine-tuned for the specific task. We provide accompanying code to reproduce all of these results.

Table 1: Test data performance of all models on different datasets.

	Resnet50	**Alexnet**	**VGG**	**Densenet**	**Squeezenet**
Adience	49.75	46.04	51.41	50.80	49.49
UTKFace	69.82	68.09	69.89	69.15	70.73
CIFAR10	83.26	92.08	89.53	85.17	76.97
CIFAR100	55.81	71.31	64.39	61.05	40.36
CIFAR100super	67.27	80.7	76.06	71.22	55.16

Figures 7, 8, 9, 10, and 11 show the measures of robustness bias as approximated by CarliniWagner (σ_P^{CW}) and DeepFool (σ_P^{DF}) across different partitions of all the datasets.

B Regularization Results

Using the regularization term derived in Section 5 (and in more detail in Appendix B.1) in the main paper, we re-train models to enure that various partitions of the dataset are equally robust, or have low robustness bias. In addition to some of the Imagenet pretrained models mentioned in Appendix A, we also investigate this on models that are trained from scratch.

As we saw in Section 4 in the main paper, Adience barely showed any robustness bias, so in this section we only focus on CIFAR-10 and UTKFace (partitioned by race). For CIFAR-10, we show an in-depth analysis of two pre-trained (and then finetuned) Imagenet models: Resnet and VGG and one Deep CNN trained from scratch, which gets an accuracy of 89% on the test set, which is comparable to the accuracy of the Imagenet models (as shown in Table 1). The Deep CNN is a popularly-used

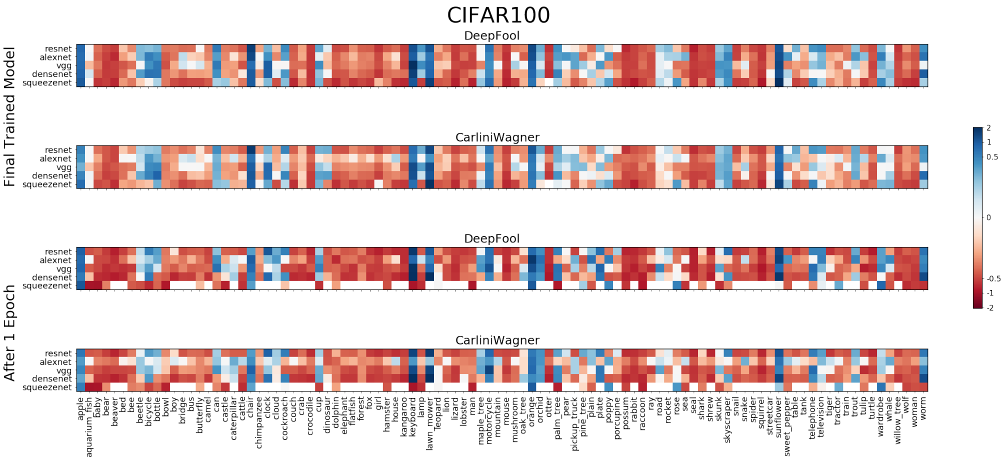


Figure 9: Depiction of σ_P^{DF} and σ_P^{CW} for the CIFAR100 dataset with partitions corresponding to the class labels $\mathcal{C}$. These values are reported for all five convolutional models both at the beginning of their training (after one epoch) and at the end. We observe that, largely, the signedness of the functions are consistent between the five models and also across the training cycle.

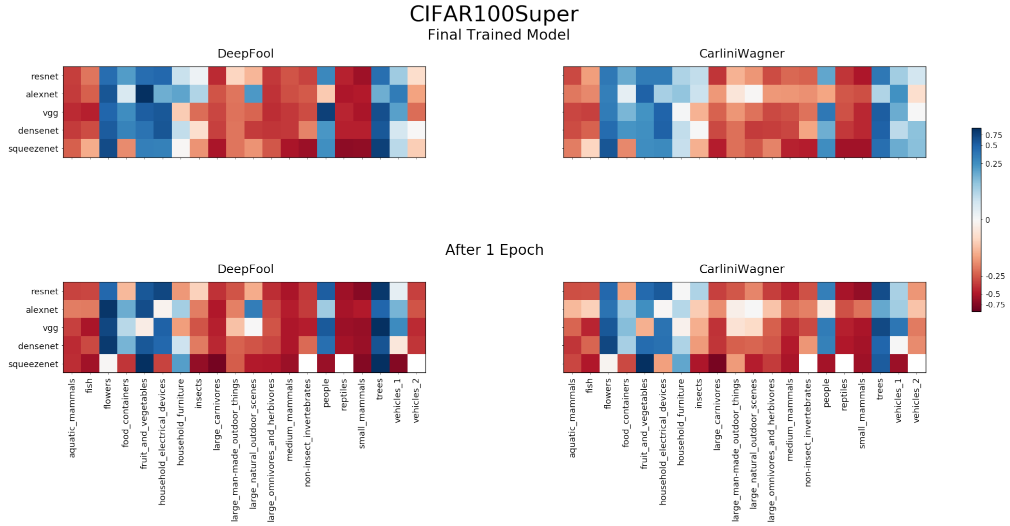


Figure 10: Depiction of σ_P^{DF} and σ_P^{CW} for the CIFAR100super dataset with partitions corresponding to the class labels $\mathcal{C}$. These values are reported for all five convolutional models both at the beginning of their training (after one epoch) and at the end. We observe that, largely, the signedness of the functions are consistent between the five models and also across the training cycle.

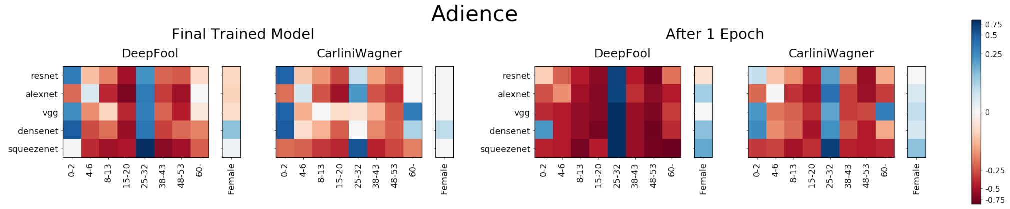


Figure 11: Depiction of σ_P^{DF} and σ_P^{CW} for the Adience dataset with partitions corresponding to the (1) class labels $\mathcal{C}$ and the and (2) gender. These values are reported for all five convolutional models both at the beginning of their training (after one epoch) and at the end. We observe that, largely, the signedness of the functions are consistent between the five models and also across the training cycle.

CNN for CIFAR-10, taken from the offical website of Torch.[3] Similarly for UTKFace, we take a

[3] http://torch.ch/blog/2015/07/30/cifar.html

publicly available CNN from Kaggle.[4] We provide PyTorch implementations of all these models in our uploaded code.

B.1 Formal Derivation of the Regularization Term

In this section, we provide a step-by-step derivation of the regularization term used in Section 5 in the main paper. Recall the traditional Empirical Risk Minimization objective, ERM $:= l_{cls}(f_\theta(X), Y)$, where l_{cls} is cross entropy loss. Now we wish to model our measure of fairness (§3) and minimize for it alongside ERM. To model our measure, we first evaluate the following cumulative distribution functions:

$$\mathbb{P}_{x \in \mathcal{D}}\{d_\theta(x) > \tau \mid x \in P, y = \hat{y}\} = \frac{1}{\sum\limits_{x \notin P} \mathbb{1}\{y = \hat{y}\}} \sum_{\substack{x \notin P \\ y = \hat{y}}} \mathbb{1}\{d_\theta(x) > \tau\}$$

$$\mathbb{P}_{x \in \mathcal{D}}\{d_\theta(x) > \tau \mid x \notin P, y = \hat{y}\} = \frac{1}{\sum\limits_{x \in P} \mathbb{1}\{y = \hat{y}\}} \sum_{\substack{x \in P \\ y = \hat{y}}} \mathbb{1}\{d_\theta(x) > \tau\}$$

This gives us the empirical estimate of the robustness bias term $RB(P, \tau)$, parameterized by partition P and threshold τ, defined as Equation 1 below.

$$\tilde{RB}(P, \tau) = \left| \frac{1}{\sum\limits_{x \notin P} \mathbb{1}\{y = \hat{y}\}} \sum_{\substack{x \notin P \\ y = \hat{y}}} \mathbb{1}\{d_\theta(x) > \tau\} - \frac{1}{\sum\limits_{x \in P} \mathbb{1}\{y = \hat{y}\}} \sum_{\substack{x \in P \\ y = \hat{y}}} \mathbb{1}\{d_\theta(x) > \tau\} \right| \tag{1}$$

Now, for $\tilde{RB}$ as defined in Equation 1 to be computed and used, for example, during training, we must approximate a closed form expression of d_θ. To formulate this, we take inspiration from the way adversarial inputs are created using DeepFool [29]. Just like in DeepFool, we also approximate distance from f_θ considering f_θ to be linear (even though it may be a highly non-linear Deep Neural Network). Thus, we get,

$$d_\theta(x) = \left| \frac{f_\theta(x)}{||\nabla_x f_\theta(x)||} \right|. \tag{2}$$

By combining Equations 1 and 2, we recover $\tilde{RB}$, a computable estimate of the robustness bias term RB, as follows.

$$\tilde{RB}(P, \tau) = \left| \frac{1}{\sum\limits_{x \notin P} \mathbb{1}\{y = \hat{y}\}} \sum_{\substack{x \notin P \\ y = \hat{y}}} \mathbb{1}\{\left| \frac{f_\theta(x)}{||\nabla_x f_\theta(x)||} \right| > \tau\} - \frac{1}{\sum\limits_{x \in P} \mathbb{1}\{y = \hat{y}\}} \sum_{\substack{x \in P \\ y = \hat{y}}} \mathbb{1}\{\left| \frac{f_\theta(x)}{||\nabla_x f_\theta(x)||} \right| > \tau\} \right|$$

Finally, given scalar α, we minimize for the new objective function, *AdvERM*, as follows:

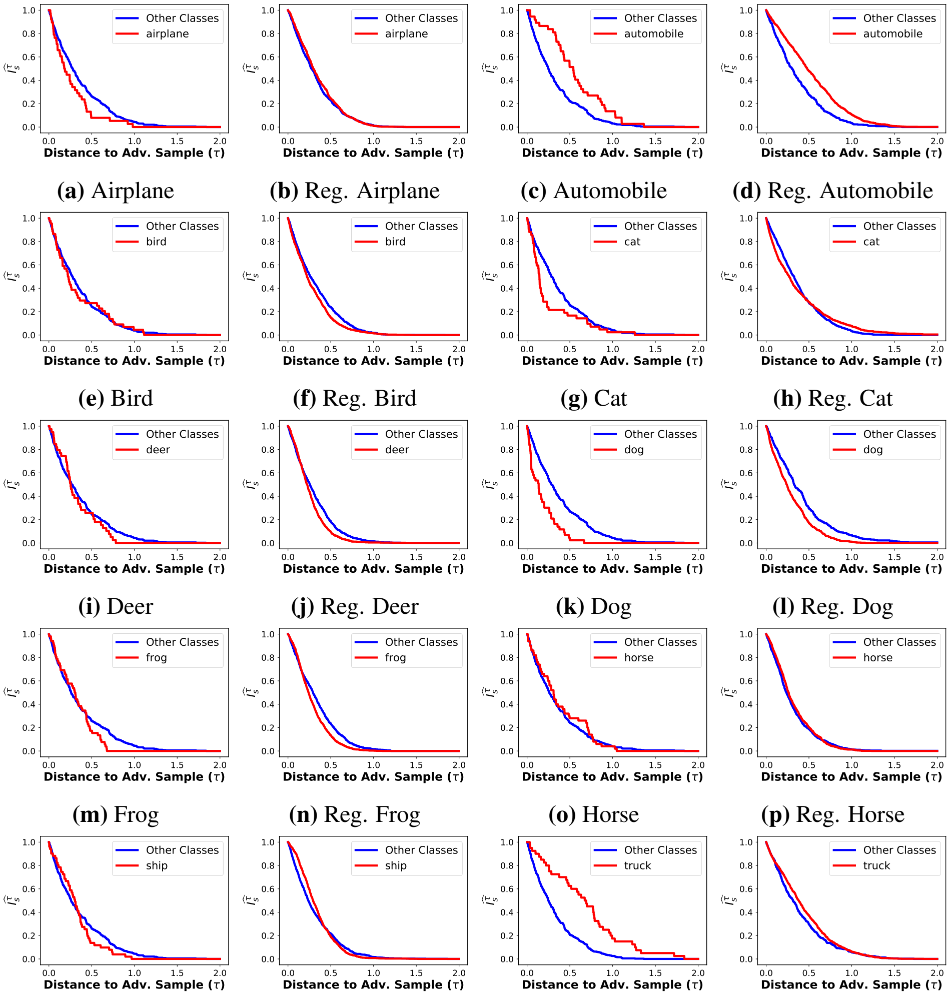
$$AdvERM := l_{cls}(f_\theta(X), Y) + \alpha \cdot \tilde{RB}(P, \tau).$$

B.2 Additional Regularization Results

Figures 12, 13, 14, 15, 16, 17 shows how models trained with our proposed regularization term show lesser robustness bias. Figures 12, 13, and 14 correspond to CIFAR-10, while Figures 15, 16, and 17

[4] https://www.kaggle.com/sangarshanan/age-group-classification-with-cnn

correspond to UTKFace. For example, for Deep CNN trained on CIFAR-10, for the partition "cat", we see that the distribution of distances become less disparate for the regularized model (Figure 12h) as compared to the original non-regularized model (Figure 12g). This trend persists across models and datasets. We provide a PyTorch implementation of the proposed regularization term in our accompanying code.



(a) Airplane **(b)** Reg. Airplane **(c)** Automobile **(d)** Reg. Automobile

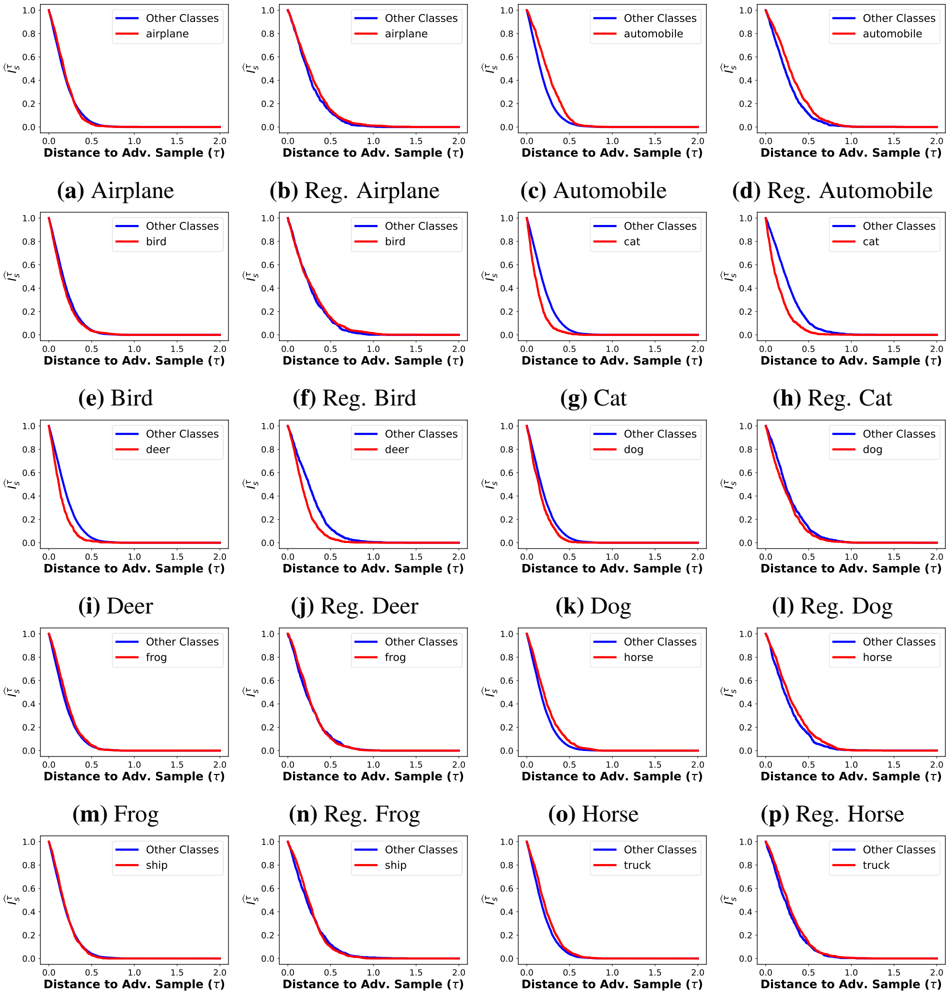
(e) Bird **(f)** Reg. Bird **(g)** Cat **(h)** Reg. Cat

(i) Deer **(j)** Reg. Deer **(k)** Dog **(l)** Reg. Dog

(m) Frog **(n)** Reg. Frog **(o)** Horse **(p)** Reg. Horse

(q) Ship **(r)** Reg. Ship **(s)** Truck **(t)** Reg. Truck

Figure 12: [Regularization] CIFAR10 - Deep CNN



(a) Airplane **(b)** Reg. Airplane **(c)** Automobile **(d)** Reg. Automobile

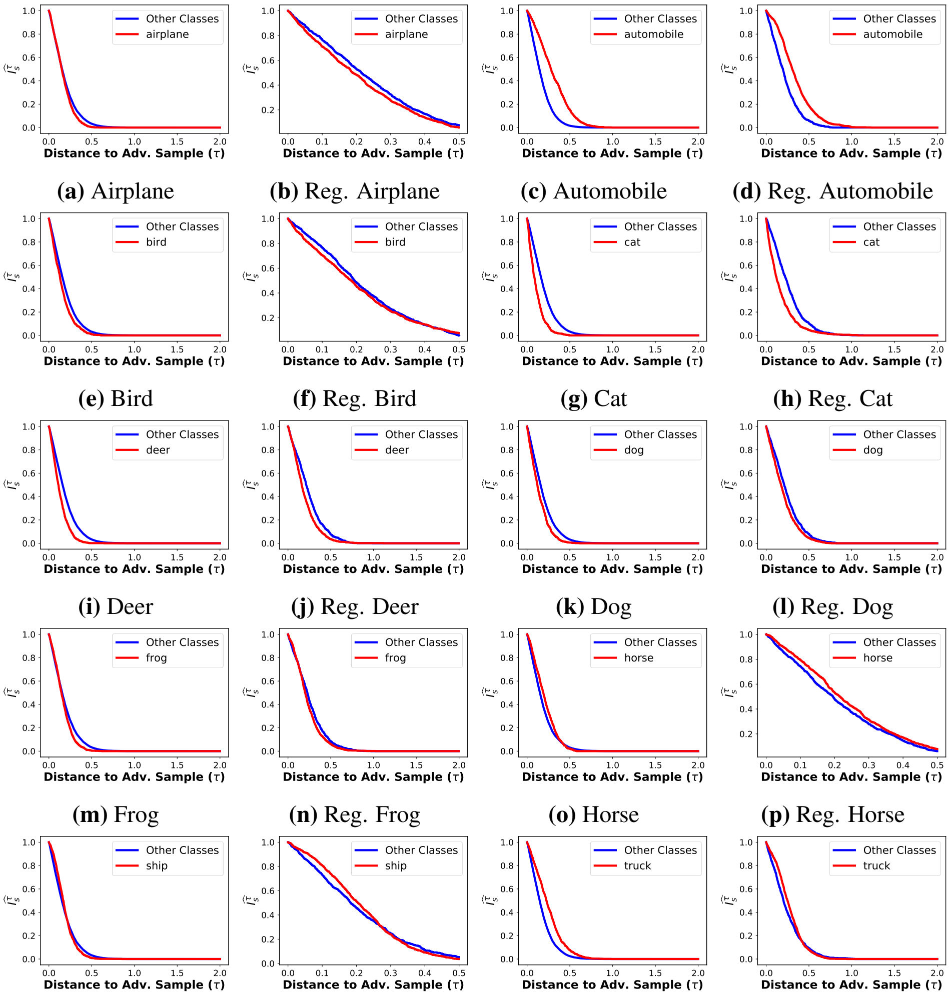
(e) Bird **(f)** Reg. Bird **(g)** Cat **(h)** Reg. Cat

(i) Deer **(j)** Reg. Deer **(k)** Dog **(l)** Reg. Dog

(m) Frog **(n)** Reg. Frog **(o)** Horse **(p)** Reg. Horse

(q) Ship **(r)** Reg. Ship **(s)** Truck **(t)** Reg. Truck

Figure 13: [Regularization] CIFAR10 - Resnet50



(a) Airplane **(b)** Reg. Airplane **(c)** Automobile **(d)** Reg. Automobile

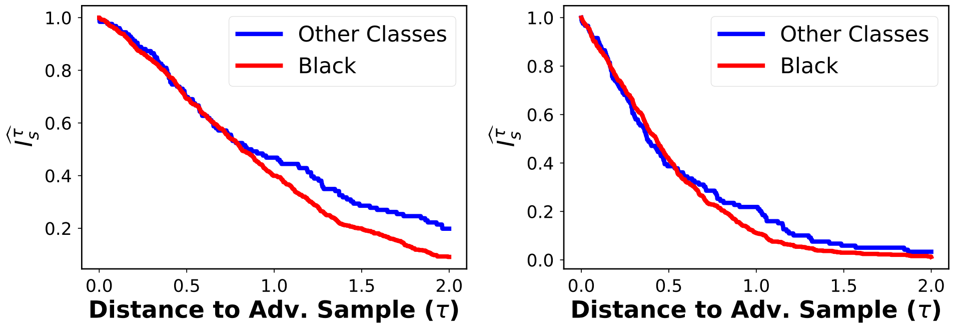
(e) Bird **(f)** Reg. Bird **(g)** Cat **(h)** Reg. Cat

(i) Deer **(j)** Reg. Deer **(k)** Dog **(l)** Reg. Dog

(m) Frog **(n)** Reg. Frog **(o)** Horse **(p)** Reg. Horse

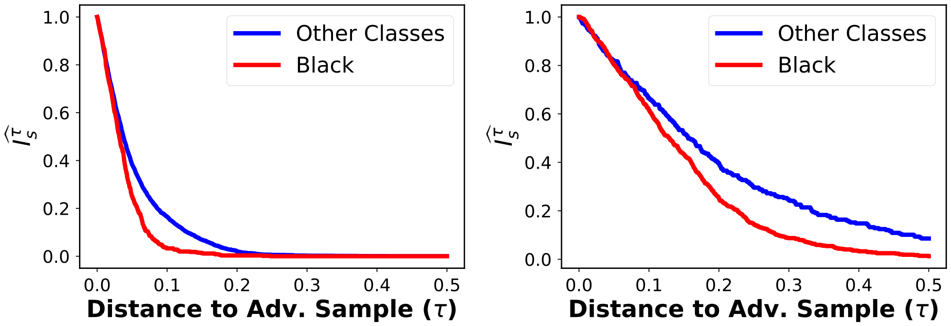
(q) Ship **(r)** Reg. Ship **(s)** Truck **(t)** Reg. Truck

Figure 14: [Regularization] CIFAR10 - VGG



(a) Partitioned by Race **(b)** Regularized for Race

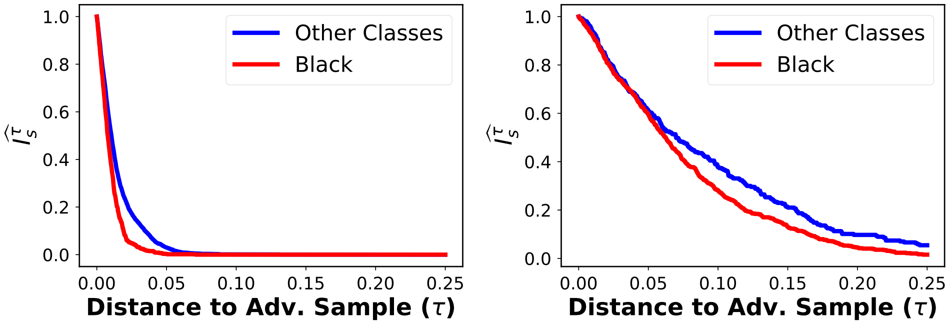
Figure 15: [Regularization] UTKFace partitioned by race - Deep CNN.



(a) Partitioned by Race

(b) Regularized for Race

Figure 16: [Regularization] UTKFace partitioned by race - Resnet50.



(a) Partitioned by Race

(b) Regularized for Race

Figure 17: [Regularization] UTKFace partitioned by race - VGG.