

The Thesis Committee for Kurtis Evan Alejo David
Certifies that this is the approved version of the following thesis:

**Debiasing Convolutional Neural Networks via
Meta Orthogonalization**

APPROVED BY

SUPERVISING COMMITTEE:

Qiang Liu, Supervisor

Raymond J. Mooney

**Debiasing Convolutional Neural Networks via
Meta Orthogonalization**

by

Kurtis Evan Alejo David

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Computer Science

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2020

Acknowledgments

Thank you Dr. Qiang Liu, for your guidance and teaching for the past 2 years. I enjoyed every discussion with you in the office; not only did I grow intellectually, but also learned effective communication skills to collaborate with other researchers.

Thank you Dr. Ruth C. Fong, for so many reasons. I am extremely grateful you offered to advise and mentor me, even if you were busy wrapping up your PhD program. This project would not have been possible without your consult and intuition. Most importantly, you were compassionate and dedicated to helping me throughout the year.

Thank you Dr. Raymond Mooney, for introducing me to the field of interpretable AI, a big inspiration for this thesis. I hope you recognize a few of the works from the interpretability meetings.

Thank you to my family and friends, for being there for me during this process, especially due to the recent pandemic. Interactions with everyone, even online, kept me going.

Thank you to my dear girlfriend Leilani, for your wonderful support through my journey in research. You taught me that it's okay to prioritize my health and take my own path to success. I greatly value your companionship and I hope you continue to listen to me rant about neural networks . . .

Debiasing Convolutional Neural Networks via Meta Orthogonalization

Kurtis Evan Alejo David, M.S.COMP.SC.
The University of Texas at Austin, 2020

Supervisor: Qiang Liu

As deep learning becomes present in many applications, we must consider possible shortcomings of these models, such as bias towards protected attributes in datasets. In this work, we focus on debiasing convolutional neural networks (CNNs), through our proposed Meta Orthogonalization algorithm. We leverage past work in debiasing word embeddings and interpretability literature to force image concepts learned by a CNN to be orthogonal to a bias direction. We empirically show through a suite of controlled bias experiments that this improves the fairness of CNNs, comparable to adversarial debiasing. We hope that this leads to new directions in debiasing and understanding deep learning models.

Table of Contents

Acknowledgments	iii
Abstract	iv
Chapter 1. Introduction	1
Chapter 2. Related Work	3
2.1 Fairness in Machine Learning	3
2.2 Adversarial Learning	5
2.3 Tangential Work on Debiasing CNNs	7
Chapter 3. Our Approach	9
3.1 Problem Formulation	9
3.1.1 Equality of Opportunity	10
3.1.2 Model Leakage	11
3.2 Debiasing Word Embeddings	11
3.3 Image Concept Embeddings	13
3.4 Measures of Class Bias	15
3.4.1 Projection Bias	15
3.4.2 Sensitivity Bias	17
3.5 Meta Orthogonalization Debiasing	18
Chapter 4. Experimental Design	21
4.1 Datasets	21
4.1.1 BAM	21
4.1.2 Bias Control in BAM	23
4.2 Adversarial Debiasing	25
4.3 Architecture and Layer Representation	25
4.4 Training and Evaluation Procedures	27

Chapter 5. Results	29
5.1 BAM: Case Study I	29
5.1.1 Equality of Opportunity	30
5.1.2 Model Leakage	31
5.1.3 Projection Bias	32
5.1.4 Sensitivity Bias	34
5.2 BAM: Case Study II	35
5.2.1 Equality of Opportunity	36
5.2.2 Model Leakage	39
5.2.3 Projection Bias	41
5.2.4 Sensitivity Bias	44
Chapter 6. Discussion	47
6.1 Future Work	48
Chapter 7. Conclusion	50
Bibliography	51

Chapter 1

Introduction

Deep learning has integrated into many applications, such as natural language understanding, autonomous driving, and medical imaging. Some of these use cases deal with high risk situations, and as such, there needs to be increased trust with these highly performant models. In addition, the data used to train these models can contain protected attributes, and often use their correlations to obtain a higher accuracy. For example, when creating models for recidivism, or the tendency for criminals to reoffend, it is the case that the datasets primarily contain certain racial profiles [3]. If followed by a naive supervised algorithm, then the model itself may exhibit racial biases, since the training dataset has skewed coverage of the sensitive information.

In another case, Bolukbasi et al. [6] revealed inherent stereotypes baked in a large corpus of human written text. Embeddings from natural language processing (NLP) such as Word2Vec [29] and GloVe [31] picked up on these stereotypes and resulted in a commonly cited analogy:

man is to *computer programmer* as *woman* is to **homemaker**

Italicized words are given as inputs to the model, and the bolded output comes from the nearest neighbor of resultant vector operations with these embeddings. This suggests two ideas: one is that our data will be naturally skewed along many dimensions, and societal biases are not an exception; and

two, we must design intelligent algorithms to prevent the model from taking advantage of these biases.

Of course, computer vision is no exception. Autonomous cars rely heavily on their vision systems, and they must also perform tasks such as pedestrian detection. A biased model in this case would be deadly. Most work in this field has focused on the reduction of fairness metrics primarily through adversarial learning [13]; however this requires careful tuning due to the inherent min-max game. We thus pose the following: can we debias convolutional neural networks (CNNs) without adversarial learning, but instead through similar geometric arguments used for word embeddings?

The nature of this paper is of an empirical exploration – using adversarial learning as a baseline, we would like to push early ideas in debiasing word embeddings to CNNs and analyze possible advantages and disadvantages. To undergo this analysis, we first provide a methodology to transfer these ideas to CNNs, *Meta Orthogonalization*. We then run comparisons on a few metrics, closely tied to concepts in fairness: Equality of Opportunity [16] and Model Leakage [36]. In addition, we motivate new measures of class specific bias, *projection bias* and *sensitivity bias* [23], and show that our debiasing framework clearly outperforms the adversarial baseline. We provide extensive analysis on multiple situations, by effectively controlling co-occurrences in the Benchmarking Attribution Methods (BAM) dataset [37].

Chapter 2

Related Work

In this section, we explore tangential work in understanding and solving model bias. We first explore the growing realm of fairness in the field, and a few of their keystone papers. As adversarial learning is our main point of comparison in this paper, we provide a brief introduction on it as well as several examples in which it is used to combat our problem. Lastly, we provide a few recent papers we believe are closely tied to the ideas within our paper.

2.1 Fairness in Machine Learning

Fairness in academic literature is closely tied with societal movements towards equal rights, such as the Civil Rights Act of 1964. During this time, work consisted of identifying unfair model predictions, including but not limited to, standardized test scores *w.r.t.* white and black demographic groups [9], as well as unfairness in employment [15]. Following suit were meta-analyses, with academia moving towards concepts of fairness (as opposed to unfairness), and outlining the disparity between notions of individual and group fairness [34]. Note that these papers are well before the recent boom in fairness in machine learning, but we can clearly see parallels with ideas being debated today. We refer [20] for a more extensive look at the parallels of past and recent history.

In the fields of fair and interpretable machine learning, a heated topic is the oversaturation of definitions of fairness; nonetheless, it is essential to have clear definitions, and understand their nuances. The first is the concept of *demographic parity* (Equation 2.1), or guaranteeing that different subgroups, often protected attributes, have equal probabilities of success in the model [39, 21, 7]

$$\Pr(\hat{Y} = 1|A = 0) = \Pr(\hat{Y} = 1|A = 1). \quad (2.1)$$

As Hardt et al. [16] highlighted, a key disadvantage to promoting this for fairness is the fact that suboptimal policies for guaranteeing equal probabilities are permitted, without the use of further constraints. Thus they define *equalized odds* and its relaxation, *equalized opportunity*. Equalized odds (Equation 2.2) adds a conditional independence restraint to demographic parity, where the prediction only has to be independent to a protected attribute given that the subpopulations both belong to the same class label (in the dataset). Simply, it aims to match the *true positive rate* and *false positive rate* of the classifier

$$\Pr(\hat{Y} = 1|A = 0, Y = y) = \Pr(\hat{Y} = 1|A = 1, Y = y), \quad y \in \{0, 1\}. \quad (2.2)$$

Equality of opportunity (Equation 2.3) instead only matches the true positive rate, i.e. allowing people who were successful, to have the same probability of succeeding, regardless of protected class

$$\Pr(\hat{Y} = 1|A = 0, Y = 1) = \Pr(\hat{Y} = 1|A = 1, Y = 1). \quad (2.3)$$

Lastly, these notions of fairness apply to subgroups, and are categorized as *group fairness constraints*, but *individual fairness* also exists. The goal is that individuals with similar features should perform similarly under a model

M [10]. This can be defined as a Lipschitz-constraint (e.g. Equation 2.4), where similarity of features and predictions are predefined by distance metrics d and D , respectively. However, there exists the choice of metric, and works such as Kim et al. [24] aim to remove the need to predefine metrics that can induce different behavior

$$D(M(x), M(x')) \leq d(x, x'). \quad (2.4)$$

Our paper will mainly focus on *group fairness*, under notions similar to equality of opportunity, but extended to the multi-class classification stage. We refer to [1] for readers who are interested in a more exhaustive list of fairness definitions.

2.2 Adversarial Learning

Adversarial learning came to the forefront by Goodfellow et al. [14], successfully training Generative Adversarial Networks (GANs). The overall goal is to train a generator G to synthesize novel samples that seem to come from a given dataset. To do this, they introduce a discriminator network D , the *adversary*, and frame the problem as a two-player min-max game (Equation 2.5). D must be learn to distinguish between a real image and a synthesized image from G , while G must learn to generate images to fool D

$$\min_G \max_D \mathbb{E}_{z \sim \mathcal{N}(0,1)} [\log(1 - D(G(z)))] + \mathbb{E}_x [\log D(x)]. \quad (2.5)$$

Much literature is dedicated to the above optimization, and have yielded high resolution, imperceptibly real images. However, for this paper, we focus on other uses of adversarial learning, i.e. introducing a second network D in hopes to induce specific behavior on the desired classification network f .

Beutel et al. [5] applies this idea to obtain demographic parity constraints. Given the output of a hidden layer in a network z , they train another network D to predict the protected class A . Lemoine et al. [26] extend this idea by enforcing orthogonal learning updates of f and D , to mitigate the leakage of information shared between the networks. They are also able to incorporate equality of odds and equality of opportunity, in addition to demographic parity constraints.

Prior work on using adversarial learning with fairness often referred to UCI datasets, but following papers have shown its utility to other more relevant fields. Wadsworth et al. [35] applies the same framework to recidivism, specifically the COMPAS dataset. They show that it can succeed and achieve state of the art results, even on high-stakes data. Madras et al. [28] introduces the notion of fair transfer learning, showing that their adversarial framework (LAFTR) can produce representations that are empirically fair on unseen tasks.

This last strategy has shown recent promise in computer vision. Wang et al. [36] highlight the gender bias in vision systems for object detection and action recognition. They utilize the same adversarial framework, but also include a learnable input-space map to visualize sections of the input their network has learned to pay less attention to. In addition, they coin a new definition, *model leakage* – the accuracy of a new model h trained to predict protected attributes, in this case gender, from the output logits of their network. As this paper most closely relates to our domain, we utilize it as our main baseline model comparison, and include their leakage metric for evaluation.

2.3 Tangential Work on Debiasing CNNs

We now highlight a few recent papers closely tied to concepts in our proposal. Chen et al. [8] analyze work on understanding image concepts through intermediate representations in the network. Specifically, they tackle the task of aligning these concepts to specific dimensions in the latent space, i.e. the learned concepts can be represented as the orthogonal standard basis vectors $e_i \in \mathbb{R}^d$. They note that without this transformation, it can be the case that post-hoc embeddings can be highly correlated, even if their corresponding concepts are not visually similar. Inspired by related whitening methods such as [19], they do this by replacing the Batch Norm layers of a network with a Concept Whitening module, that learns the optimal whitening transformation to align concepts to specific dimensions in the latent space. As the authors have noted, however, this may not be optimal for every set of concepts, because there can exist strong correlations, like *cloud* and *sky* concepts. Our work is unique from their method in that we only make concepts orthogonal to a defined bias subspace, rather than all pairs. Our method also does not require the addition of new parameters because of whitening.

Another aspect we would like to touch on is meta-learning; although our primary goal is not to generalize to several tasks given limited data [11, 2], we use similar inner optimization steps to debias CNNs. This is more in line with work such as [32], that aim to decorrelate class-specific saliency maps. Normally, saliency maps utilize first order gradient terms [33], but instead they first take a gradient descent step to improve the loss of the minimum class, and then compute image gradients with this meta-network. Our method is similar, in that we utilize second order gradients; however, we compute a meta-step on auxiliary classifiers (and not on the CNN) to produce our penalty loss term.

Lastly, a major assumption of our dataset in the paper is that bias in CNNs arises through object co-occurrences. This is in line with works such as Lapuschkin et al. [25] – they find that major datasets contained photos that were tagged by reoccurring watermarks. Through saliency methods, they show that the neural network heavily weighted those pixels, and when removed from these images, the network would misclassify. Because these watermarks do not have explicit labels, they proposed an unsupervised clustering method on the saliency maps to automatically detect various learning behaviors of the CNN, including the dependence of the logos. Similarly, Yang and Kim [37] found that under a 1:1 co-occurrence between a pasted object and scene, the saliency maps would indicate that the CNN would pay less attention to the object, compared if only a small ratio of images had these pasted objects. We explore this further in Chapter 4, as we extensively use their proposed dataset BAM.

Chapter 3

Our Approach

In this chapter, we open with our problem formulation, stating measures of fairness, *equality of opportunity* and *model leakage*, we would like our CNN to have. We first motivate our approach by providing relevant work in NLP, mainly debiasing word embeddings. We then provide the CNN counterpart, *image concept embeddings* – through this we introduce and motivate two additional measures of concept bias, namely *projection bias* and *sensitivity bias*, derived from the work in word embeddings and interpretability, respectively. Finally, we introduce our algorithm *Meta Orthogonalization* that aims to do well under these measures of concept bias, and through reformulations, show why it could also do well in maximizing fairness.

3.1 Problem Formulation

We now formalize the task explored in the rest of the paper. Let $\mathcal{D} = \{(x^{(i)}, Y^{(i)}, A^{(i)})\}$ be our dataset of images $x^{(i)} \in \mathbb{R}^{3 \times H \times W}$, labels $Y^{(i)} \in \{0, 1, \dots, N - 1\}$, and protected attributes $A^{(i)} \in \{0, 1\}$ ¹. Our downstream task is image classification – being able to predict $Y^{(i)}$ given $x^{(i)}$. To do this, we learn a CNN $f_\theta : \mathbb{R}^{3 \times H \times W} \rightarrow \mathbb{R}^N$, and given a desired layer l ,

¹In this work, we only consider binary labels for protected attributes, but recognize that this may not be representative of all possible values, e.g. gender, race, sexuality.

$f_\theta^{(l)} : \mathbb{R}^{3 \times H \times W} \rightarrow \mathbb{R}^d$ will denote its intermediate representation at layer l . In *standard training*, we would like to learn a set of parameters θ for our CNN f_θ to minimize classification loss, namely cross entropy, over the training set. We will refer to this as $\mathcal{L}_{\text{class}}(\theta)$.

In this work, we go further and would like our CNN to be “fair” *w.r.t.* the bias attribute A . To measure fairness, we evaluate on the following two metrics: Equality of Opportunity and Model Leakage.

3.1.1 Equality of Opportunity

Our first metric first measures the true positive rate of a specific class, given the protected attribute $a \in \{0, 1\}$. Specifically, given specific class $y = 0, \dots, N - 1$, we can compute its approximation according to a dataset:

$$\Pr(\hat{Y} = y | A = a, Y = y) \approx \frac{1}{|S_{a,y}|} \sum_{(x,A,Y) \in S_{a,y}} \mathbb{1}[f(x) = y] \quad (3.1)$$

where $S_{a,y} = \{(x^{(i)}, A^{(i)}, Y^{(i)}) \in \mathcal{D} : A^{(i)} = a, Y^{(i)} = y\}$

Because our task has multiple classes, we define a set of binary random variables $Z_y = \begin{cases} 1 & \text{if } Y = y \\ 0 & \text{otherwise} \end{cases}$, corresponding to each class y . Similarly, for predictions \hat{Y} , we can define $\hat{Z}_y = \begin{cases} 1 & \text{if } \hat{Y} = y \\ 0 & \text{otherwise} \end{cases}$. From this reformulation, we can exactly define *equality of opportunity*, specific to each class y :

$$\Pr(\hat{Z}_y = 1 | A = 0, Z_y = 1) = \Pr(\hat{Z}_y = 1 | A = 1, Z_y = 1) \quad (3.2)$$

To measure each model’s faithfulness to this equality, we compute the absolute difference of both sides in Equation 3.2 using the approximation in Equation 3.1. We denote this as the Equality of Opportunity Discrepancy, and is our first metric.

3.1.2 Model Leakage

Model leakage was first proposed by Wang et al. [36], which measures the accuracy of another model h trained to predict protected attributes A , using the final logits of the network:

$$\Lambda(f_\theta) = \frac{1}{|\mathcal{D}|} \sum_{i=0}^{|\mathcal{D}|} \mathbb{1}[h(f(x^{(i)})) = A^{(i)}] \quad (3.3)$$

This is an interesting measure, because these logits are exactly what is used to make the final prediction, through an argmax operation. Thus, this can also measure a degree of fairness of the network; specifically, the amount of predictive information on the protected attribute the final outputs contain. Ideally we want leakage to be as close to 50% as possible. However, Wang et al. [36] note that Equation 3.3 is lower bounded by the dataset leakage, a similar measure, but using $Y^{(i)}$ as the inputs to h . We omit taking this measure, as comparisons between models will not be affected by this value.

3.2 Debiasing Word Embeddings

Before introducing our method, we motivate it through literature in NLP. Throughout the paper, we will be studying *embeddings*, or vectors in \mathbb{R}^d that represent some entity. Word embeddings such as Word2Vec [29], map words in a vocabulary \mathcal{V} to corresponding vectors. These vectors are learned parameters – in the case of Word2Vec, they are optimized to maximize skip-gram performance, i.e. given a word in a sentence, predict the other (context) words in the sentence. Although completely unsupervised, these embeddings have been shown to perform well when used for other tasks.

Another interesting property of these word embeddings is that they

contain semantic meaning *w.r.t.* vector addition. This allows us to solve analogies in the form:

A is to B as C is to ___ ?

Specifically, let $\beta_A, \beta_B, \beta_C \in \mathbb{R}^d$ be the embeddings of words A , B , and C respectively. In vector representation, we have that our target vector $\hat{\beta}$ should follow $\hat{\beta} - \beta_C \approx \beta_B - \beta_A$. Using this, then our target word D can be found by a nearest neighbor optimization:

$$D = \operatorname{argmin}_{w \in \mathcal{V}} \|\beta_w - \hat{\beta}\|_2. \quad (3.4)$$

As noted in the introduction, however, Bolukbasi et al. [6] have shown that these analogy completions can have severe bias. For example, given non-gender related words such as professions, the model would bias towards a certain gender more heavily. To solve this, first note that Equation 3.4 can be rewritten using a *similarity metric*, such as cosine similarity. This directly corresponds to the standard inner product of \mathbb{R}^d , so that we can now solve instead:

$$D = \operatorname{argmax}_{w \in \mathcal{V}} \frac{\beta_w^\top \hat{\beta}}{\|\beta_w\|_2 \|\hat{\beta}\|_2}. \quad (3.5)$$

This provides insight to their following solution; without loss of generality, they first assume binary gender, and learn an overall *gender direction*. Simply, given a pair of sets of gender-specific word embeddings \mathcal{B}_M and \mathcal{B}_F , they compute a 1-dimensional subspace for each using the highest principal vector β_M and β_F , respectively. Their overall gender direction is now provided by

$$\nu \triangleq \beta_M - \beta_F.$$

With this, given a set of word embeddings \mathcal{W} to be debiased, they perform a simple projection to debias vectors, under a constant to keep vectors normalized. Specifically, let $\beta \in \mathcal{W}$. Its biased representation under the one-dimensional subspace is exactly $\beta_{\mathcal{B}} = (\beta^{\top} \nu) \nu$; and thus its *hard debiased* form is computed by its *orthogonal projection* to \mathcal{B} , or $\beta_{\perp} = \beta - \beta_{\mathcal{B}}$. They note that this can be too strong of a linear transformation, and lose out on semantic meaning, and thus they also introduce *soft debiasing*, which allows them to optimize for a transformation to induce orthogonality of gender subspaces in addition to retaining semantic meaning [6].

With regards to our paper, we would like to learn debiased representations from the ground up, i.e. apply some type of constraint during training. Kaneko and Bollegala [22] opt to do this by adding a penalty, or regularization, term when learning GloVe embeddings:

$$\mathcal{L}_{\text{debias}}(\beta) = \sum_{w \in \mathcal{V}} \left(\frac{\beta_w^{\top} \nu}{\|\beta_w\|_2 \|\nu\|_2} \right)^2. \quad (3.6)$$

An advantage to this is that due to the natural trade-off given by a regularization term, this allows for embeddings to retain semantic meaning and orthogonality to bias, without having to restrict to using a linear transformation as a post-processing step. We utilize this exact loss, but first provide how to extract analogous image concept embeddings.

3.3 Image Concept Embeddings

In a similar fashion, *concepts* in images (such as colors, objects, textures) can also be represented as embedding vectors. Given a trained CNN f_{θ} , and dataset \mathcal{D} , suppose we have auxiliary concept labels. Given concept c , we would like to learn its corresponding embedding β_c . To do this, Fong and

Vedaldi [12], Kim et al. [23] learn linear classifiers $g_c = \beta_c^\top z + b_c$ that operate on some intermediate representation of the network, i.e. $z_\theta(x) = f_\theta^{(l)}(x)$ ². Specifically, they partition \mathcal{D} into concept-specific datasets, and learn linear hyperplanes by minimizing a binary log-loss:

$$\begin{aligned} \mathcal{L}_{\text{concept}}(c, \beta, b) = & - \sum_{(x^{(i)}, c^{(i)})} c^{(i)} \log(\sigma(g_c(z_\theta(x^{(i)}))) \\ & + (1 - c^{(i)}) \log(1 - \sigma(g_c(z_\theta(x^{(i)})))) \end{aligned} \quad (3.7)$$

which learns to separate representations of images that contain the concept c , and the images that do not.

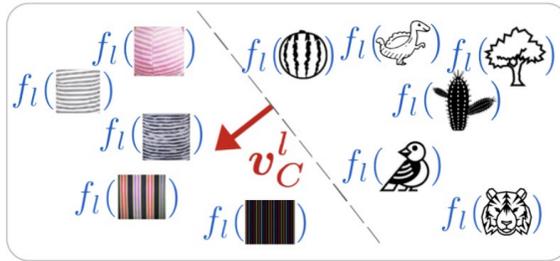


Figure 3.1: A visual of a learned image concept, from the original paper [23]. v_C^l is equivalent to the learned β_c .

Once Equation 3.7 is minimized, we obtain learned β_c that directly point to examples that are positive for concept c . See Figure 3.1 as a visual. Finally, Fong and Vedaldi [12] observe that these concepts also have similar vector arithmetic properties as word embeddings; thus this might suggest that we can also debias image concept embeddings through similar geometric arguments such as in Equation 3.6.

²In the case that layer l is convolutional, i.e. its activation is a 3D tensor, the activation is first spatially summed to produce a vector.

3.4 Measures of Class Bias

We can now merge ideas from Section 3.2 and 3.3, to define additional measures of bias in a CNN. According to Section 3.3, given concept labels, we can learn image concept embeddings. Initially this may not seem directly connected to debiasing the predictions of our CNN, but notice that the *image labels* $\{Y^{(i)}\}$ can also act as concept labels, where each class $y = 0, \dots, N - 1$ can also be our concepts $c = 0, \dots, N - 1$. Although these class embeddings are not as predictive as the final predictions of the CNN, these embeddings provide how much information a specific layer l has already extracted. If this is sufficient for linear classification, then we should get highly informative vectors. In addition, our protected label $\{A^{(i)}\}$ will also be represented as a concept embedding. To match with Bolukbasi et al. [6], we actually learn **two** separate embeddings for our protected concept, one for positive examples (β_{A^+}) and one for negative examples (β_{A^-}). This allows us to define our bias direction:

$$\nu_{\text{bias}} \triangleq \beta_{A^+} - \beta_{A^-}. \quad (3.8)$$

3.4.1 Projection Bias

Using Equation 3.8, and learned class embeddings β_c , we can measure the class-specific projection bias of the CNN:

$$\omega(c, \beta) = \frac{\beta_c^\top \nu_{\text{bias}}}{\|\beta_c\|_2 \|\nu_{\text{bias}}\|_2}. \quad (3.9)$$

This is simply the cosine similarity of the class embeddings *w.r.t.* the bias direction. If this is non-zero, then that means that the CNN has learned to correlate class-specific information with predictive information of the protected attribute. Ideally this should be zero, i.e. the class embeddings are

orthogonal to the learned bias. Because this directly connects with class labels that are also associated with the final output layer, we hope that Equation 3.9 is correlated with our desired measures of fairness. This intuition also provides the basis for our proposed method; see Figure 3.2 for the visual. We provide the details in Section 3.5.

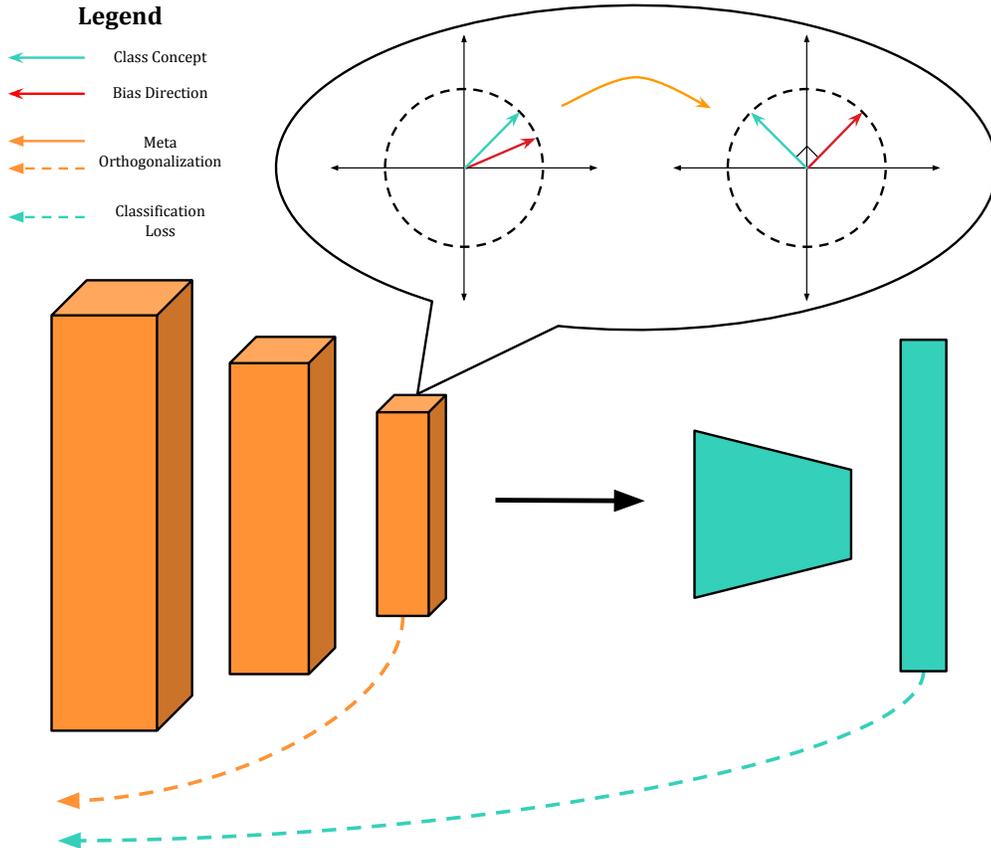


Figure 3.2: Visualization of our method. The goal is to update the convolutional parameters (in orange) such that class embeddings in the latent space are orthogonal to the learned bias direction. Dashed lines represent gradients used during backpropagation.

3.4.2 Sensitivity Bias

Additionally, we would like to motivate a new method of assessing bias in a network. Kim et al. [23] go further and describe a way to determine which concepts are most responsible for the final prediction. Although we do not use their proposed TCAV metric, we are interested in a value that they compute: the *directional derivative*. This scalar that quantifies the rate at which a function changes in a certain direction u . More formally, it is defined as:

$$\begin{aligned} \lim_{h \rightarrow 0} \frac{f(\mathbf{x} + h\mathbf{u}) - f(\mathbf{x})}{h} &= \nabla_{\mathbf{u}} f(\mathbf{x}) \\ &= \nabla f(\mathbf{x}) \cdot \mathbf{u}. \end{aligned} \tag{3.10}$$

We can see that this is exactly measuring the rate of change of a function in a given direction u . If f is a particular logit of the CNN, Kim et al. [23] uses the directional derivative in addition to a minimum threshold on this scalar value to determine which outputs are significantly affected by certain concepts. This directly relates to our discussion of bias, because from Equation 3.8, we can set $u = \nu_{\text{bias}}$, or the bias direction, and $\nabla f(x)$ can be computed for any of our output classes. This setup directly defines our notion of *sensitivity bias*.

Notice that this is in close relation to the previous projection bias. The only difference is that instead of learning class-specific embeddings in the latent space, we can use the gradient of the output prediction *w.r.t.* the intermediate layer l ; thus we can provide a measure of bias on the *input* level, rather than a holistic dataset version. Intuitively, sensitivity bias answers the following question: “How much will the current prediction increase in confidence if we increase the amount of predictive information of bias in the intermediate representation?” To our knowledge, no other paper has used this

metric in the study of bias; however similar arguments of sensitivity have been shown in [4, 30] for visual interpretability in neural networks.

3.5 Meta Orthogonalization Debiasing

Motivated by the work in the past few sections, we now provide specifics on our algorithm. First, we augment the training process by also learning image concept embeddings simultaneously with our CNN f_θ . As stated, the concepts of interest are exactly the N classes in the image classification task, in addition to A^+ and A^- , the positive and negative denotions of the protected attribute. These image concepts are treated as a multi-label task setting, i.e. we simultaneously minimize $\mathcal{L}_{\text{concept}}(c, \beta, b)$ (Equation 3.7) for every stated concept c . Lastly, following our intuition on the projection bias, we use Equation 3.9 as a constraint on our learned class embeddings, so that they are orthogonal. In total we have the following optimization:

$$\begin{aligned} \min_{\theta, \beta, b} \quad & \mathcal{L}_{\text{class}}(\theta) + \sum_c \mathcal{L}_{\text{concept}}(c, \beta, b) \\ \text{s.t.} \quad & \omega(c, \beta) = 0 \quad c = 0, \dots, N - 1. \end{aligned} \tag{3.11}$$

We could approximately solve Optimization 3.11 through SGD, in addition to a projection step for the β parameters to satisfy the constraint. As noted by Bolukbasi et al. [6], this projection can remove useful semantic information; therefore we can instead approximate this constraint with the penalty defined in Equation 3.6:

$$\min_{\theta, \beta, b} \quad \mathcal{L}_{\text{class}}(\theta) + \sum_c \mathcal{L}_{\text{concept}}(c, \beta, b) + \lambda \mathcal{L}_{\text{debias}}(\beta). \tag{3.12}$$

This, however, still does not affect our parameters θ , and thus has no effect on the bias of our CNN. It is the case, though, that the image

concept embeddings β are updated at every minibatch. Assuming that these are learned using gradient descent with learning rate α , we can define a specific concept embedding update like so:

$$\beta'_c \triangleq \beta_c - \alpha \nabla_{\beta_c} \mathcal{L}_{\text{concept}}(c, \beta, b). \quad (3.13)$$

This is crucial, as $\mathcal{L}_{\text{concept}}$ at every minibatch is a function of θ ; recall Equation 3.7, where the log-loss is defined on classifying intermediate representations $z_\theta(x^{(i)})$. Therefore, to induce a change in θ through the debias loss, we instead take a *meta-step* on our embeddings β defined in Equation 3.13, so that we can backpropagate the loss to the convolutional parameters θ :

$$\min_{\theta, \beta, b} \mathcal{L}_{\text{class}}(\theta) + \sum_c \mathcal{L}_{\text{concept}}(c, \beta, b) + \lambda \mathcal{L}_{\text{debias}}(\beta'). \quad (3.14)$$

Essentially, what this does is condition our CNN so that more accurate concept embeddings (i.e. if we are to take an additional update step) are naturally orthogonal in the latent space. We provide a practical algorithm run per epoch for training in Algorithm 1.

Algorithm 1 Meta Orthogonalization

Require: CNN f_θ , linear classifiers $g_c(\beta, b)$, dataset \mathcal{D} , regularization parameter λ , concept learning rate α

```
1: function TRAIN-EPOCH( $f_\theta, \beta, \mathcal{D}, \lambda$ )
2:   for minibatch  $M \in \mathcal{D}$  do
3:     loss  $\leftarrow \mathcal{L}_{\text{class}}(f_\theta, M)$ 
4:     for concept  $c$  do
5:        $M_c \leftarrow$  sample  $M$  s.t. 50-50 ratio in examples with concept  $c$ 
6:       loss  $\leftarrow$  loss +  $\mathcal{L}_{\text{concept}}(c, \beta, b, M_c)$ 
7:        $\beta'_c \leftarrow \beta_c - \alpha \nabla_{\beta_c} \mathcal{L}_{\text{concept}}(c, \beta, b, M_c)$ 
8:     end for
9:     loss  $\leftarrow$  loss +  $\lambda \mathcal{L}_{\text{debias}}(\beta')$ 
10:    Backpropagate loss w.r.t.  $\theta$  and  $\beta, b$ 
11:  end for
12:  return  $f_\theta$ 
13: end function
```

Chapter 4

Experimental Design

Another aspect of interest in this project is how to perform rigorous studies of bias between the two methods. This section is dedicated to describing dataset and design decisions, so that we can study a variety of situations where bias can affect training of CNNs. We also include settings used for training and evaluation for these experiments to help with reproducibility of these tests.

4.1 Datasets

4.1.1 BAM

Yang and Kim [37] provides a new dataset, the Benchmarking Attribution Methods (BAM) dataset, as a way to be able to control object co-occurrence statistics and observe the behavior of proposed interpretability methods. Specifically, the whole dataset is a mixture of cropped objects from MSCOCO [27] and scenes MiniPlaces [38]. Their methodology allows a user to make their own dataset, with certain objects pasted in specific scenes. In their paper, they use this to study which types of features classification networks utilize in the input image. For example, they can train a network that contains a dog in 50% of the images, and another network that contains a dog in all of the images. They then compare the accuracy of these two networks when applied to images without any dogs – interestingly, they find that the network

trained with dogs in all images perform much significantly better, because it has learned to effectively ignore the dog objects.

The ability to control the % of objects pasted in the scene images is the core of our experimental design. In all of our tests, we define the protected attribute as the occurrence of a certain object – does the image contain a truck or a zebra? Thus, as described in Section 3.4, we learn image concept embeddings for detecting a truck and for detecting a zebra in the input images. The resultant bias direction is just the difference between these two embeddings. We use all 10 scenes from the dataset, and train the CNNs to classify scenes. We now use the next subsection to describe how we construct various datasets of varying bias ratios.

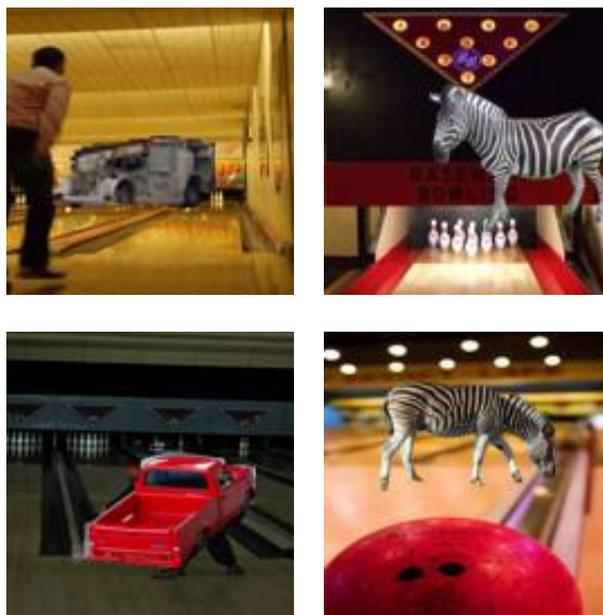


Figure 4.1: Example images from BAM, in the class “bowling alley.” **Left:** Images pasted with “truck” objects. **Right:** Images pasted with “zebra” objects.

4.1.2 Bias Control in BAM

Our goal is to be able to observe trends in our metrics *w.r.t.* a controllable bias parameter. To do this, we introduce a ratio $\rho_s \in [0, 1]$, representing the % of images in a specific scene class s that have a randomly cropped truck in them; the remaining $1 - \rho_s$ images of scene s will have a randomly cropped zebra. Thus, ρ_s can be thought of as just the co-occurrence of the truck object with scene s . We now describe how we use ρ_s to induce biased networks.

In the ideal case, we hope that training datasets have $\rho_s = 0.5$ for every class. To deviate from this setting, we choose a specific **biased** subset $\mathcal{K} \subseteq [N]$ of classes such that they have a new ratio $\rho_{\mathcal{K}}$. Any classes not in this subset retain a ratio of 0.5. Our hope is that the network will exhibit biased behavior in this subset of images. To verify this, we train CNNs on datasets where $\mathcal{K} = \{\text{bowling_alley}\}$ and vary $\rho_{\mathcal{K}}$. We then learn an image concept embedding for the single scene in \mathcal{K} , as well as the truck and zebra concepts, and measure its resultant projection on the bias direction, $\nu_{\text{bias}} = x_{\text{truck}} - x_{\text{zebra}}$.

Figure 4.2 shows the relationship between the projection and varying ratio. Note the strong positive correlation of $\rho_{\mathcal{K}}$. This is exactly what we expect, since a higher projection indicates that the bowling alley concept embedding is more similar to the trucks (and this happens when the co-occurrence is high). Similarly, a more negative projection indicates that the bowling alley concept embedding is more similar to the zebra embedding.

With the introduction of subset \mathcal{K} , we end up with another factor to control. Specifically, we study different cardinalities of \mathcal{K} : $\{1, 3, 5, 7, 10\}$. We randomly choose which labels to be in \mathcal{K} . We split our analysis into the following two cases:

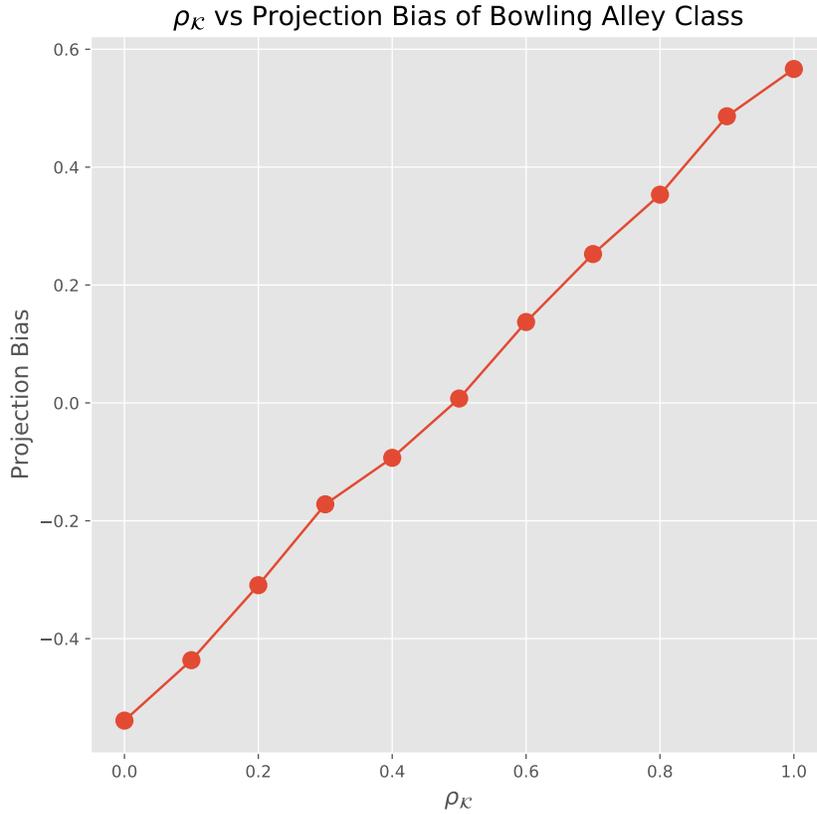


Figure 4.2: Effect of $\rho_{\mathcal{K}}$ on bias projection of the bowling alley concept.

1. **Case Study I:** $|\mathcal{K}| = 1$. This is the simplest case, where only a single class has a controllable bias ratio. This allows us to have an initial gauge of effectiveness of our debiasing framework.
2. **Case Study II:** $|\mathcal{K}| > 1$. Because this case includes multiple classes, our goal is to see how effective our method can be as the amount of bias increases overall in the dataset.

Note that in each of these cases, every class within the set will have the same assigned $\rho_{\mathcal{K}}$; this allows us to systematically control $\rho_{\mathcal{K}}$, since the space

of possible ratios would exponentially increase as $|\mathcal{K}|$ increases. For all results shown in Chapter 5, we use $\rho_{\mathcal{K}} = \{0, 0.25, 0.5, 0.75, 1.0\}$ for each Case Study.

4.2 Adversarial Debiasing

As our main baseline to compare to, we briefly describe adversarial debiasing. As discussed in Section 2.2, the goal of this framework is to use another network, say $h : \mathbb{R}^d \rightarrow \mathbb{R}$ that detects for the protected attribute, given an intermediate layer’s activations. We then train our CNN f_{θ} so that the accuracy of h declines – forcing the intermediate representation to remove information useful to h . Formally, we perform a min-max game with an extra loss term on the loss function of h :

$$\min_{\theta} \max_h \mathcal{L}_{\text{class}}(f_{\theta}) - \lambda_{\text{adv}} \mathcal{L}_{\text{protected}}(h). \quad (4.1)$$

λ then controls the strength of regularization, and often controls the speed at which h obtains near 50% accuracy. Across all experiments, we use $\lambda_{\text{adv}} = 0.5$. We find that pretraining h is necessary for 4.1 to converge in practice.

4.3 Architecture and Layer Representation

In this study, we choose ResNet-50 [17] as our CNN. We found it to have strong enough capacity for the tasks, achieving about 90% accuracy on our tasks without significant tuning or data augmentations. However, because the methods being tested are entirely dependent on a specific intermediate layer, running the above procedure with the depth of a ResNet easily becomes a large search space. Therefore, we apply the following heuristic to pre-emptively choose which layer to debias at.

We first train the CNNs using standard training, and learn the desired image concept embeddings at each possible layer. Because one of our main metrics of interest is *sensitivity bias*, we measure the varying amounts of this bias at each layer. We plot the resultant amounts of bias in Figures 4.3.

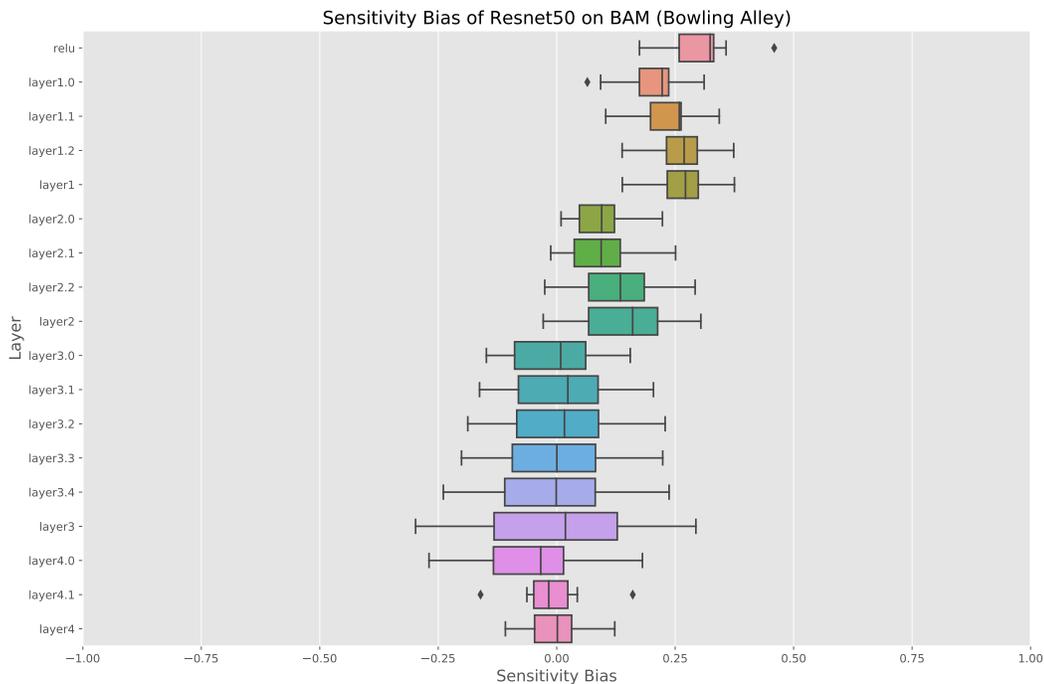


Figure 4.3: Sensitivity bias at every layer in ResNet-50 for BAM, specific to the bowling alley class.

We see that for BAM, the widest breadth of values occurs in “layer3¹.” This means that across all situations we would like to study, these layers have the highest variance of possible values of our sensitivity bias metric;

¹These names refer to instance variables corresponding to official implementations of ResNet in PyTorch. They can be found here: <https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py>

therefore, we would like to study the effect of different methods on this layer, i.e. both Meta Orthogonalization Debiasing and Adversarial Debiasing use this representation as inputs to their respective classifiers, for every experimental dataset.

4.4 Training and Evaluation Procedures

We now describe the specifics of training to make Meta Orthogonalization Debiasing possible. First, we finetune ResNet-50 pretrained on ImageNet, to the desired training dataset for 30 epochs. We use SGD as our optimizer with learning rate 0.01 and momentum 0.9. Afterwards, we begin training of image concept embeddings concurrently with the proposed meta-loss regularization as described in Section 3.5. The learning rate is then reduced to 0.001 for BAM, and we do this for 15 additional epochs. Because the learning of embeddings β and parameters θ is independent, we use a separate SGD optimizer for β , with learning rate 0.05, and no momentum. In addition, because it can be costly to train each embedding separately, we parallelize learning by using the same stream of batches sent to train the CNN. To balance the batches, we ensure that each embedding sees a 50% split on the concept they are being trained for. Our training batch size is 64, and finally we set $\lambda_{\text{debias}} = 1000$ for all tests of our proposed method.

For adversarial debiasing, we perform the same first step, but during the second step, we apply an adversarial loss instead with $\lambda_{\text{adv}} = 0.5$. The adversary is trained with learning rate 0.05, and we monitor the loss of the adversary. Because the regularization is designed to have a high loss, we allow the adversary to retrain once it reaches a set threshold for its loss. In addition, like Wang et al. [36], we find that pretraining h before using it in the adversarial

loss is essential to efficiently train and converge across mini-batches.

All of our metrics, with the exception of equality of opportunity, require an additional trained model on top of the learned parameters θ . The most straightforward is model leakage, where we learn a standard multilayer perceptron, taking inputs from the specific convolutional layer of interest, and predicting for the protected attribute. This is trained with an Adam optimizer with learning rate .001 for 20 epochs.

For the other two metrics, however, image concept embeddings are needed. One can argue that for our method, we can use the trained embeddings, and we would definitely have nearly orthogonal projections to the bias direction if the regularization is performed correctly. This is similar to reporting accuracy of a classifier on a training set, so instead, we randomly initialize *new embeddings* and retrain them. The hope is that even if we learn new image concept embeddings, the class embeddings will remain orthogonal to the bias direction. We do the same for our sensitivity bias metric, i.e. learn new embeddings to compute the bias direction. However, we note that because this is an instance-level metric, we actually report the average sensitivity bias per class:

$$\mathcal{S}(\theta, c, \nu_{\text{bias}}) = \mathbb{E}_{\{(x,y) : y=c\}} \frac{\nabla_z f_{\theta}(x)^{\top} \nu_{\text{bias}}}{\|\nabla_z f_{\theta}(x)\|_2 \|\nu_{\text{bias}}\|_2}. \quad (4.2)$$

These also work for the adversarially trained model, because we keep the learned parameters θ constant, and only need to obtain post-hoc image concepts from the network. Lastly, for every metric, we use a test set that has $\rho_{\mathcal{X}} = 0.5$, i.e. all classes have little co-occurrence with the protected attribute. We want to measure the worst-case scenario for a biased network – an unbiased population and see the extent to which its bias shows in the metrics.

Chapter 5

Results

We now compare (1) Standard Training, (2) Meta Orthogonalization, and (3) Adversarial Debiasing across our evaluation metrics. Our goal is to study how varying the bias ratio $\rho_{\mathcal{K}}$ affects these models. Because the size of the biased set, $|\mathcal{K}|$, will vary, we split the analysis into two Case Studies; Case Study I will focus on $|\mathcal{K}| = 1$, specifically biasing the “bowling alley” class, while Case Study II will encompass $|\mathcal{K}| > 1$. In each Case Study, we study the effect on the evaluation metrics in both the **biased set** \mathcal{K} as well as the **unbiased set** \mathcal{K}^c . In addition, every result is accompanied by standard error bars (denoted by the colored areas around the lines). This comes from doing 3 trials for each model.

5.1 BAM: Case Study I

In this study we generate training datasets such that a single class, “bowling alley”, will have a varying bias ratio, while the other classes (\mathcal{K}^c) are kept at a constant bias ratio of 0.5. Because the unbiased set encompasses nine other classes, we provide average results for our figures.

5.1.1 Equality of Opportunity

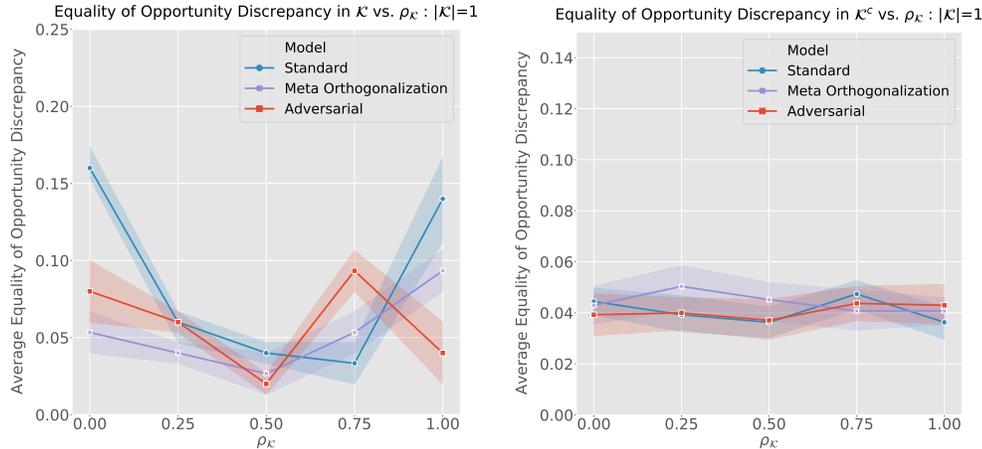


Figure 5.1: Comparing the opportunity discrepancy of models across varying $\rho_{\mathcal{K}}$. **Left:** Specific to the bowling alley class. **Right:** Averaged over all classes not included in \mathcal{K} . Notice the clear parabolic trend vs. flat line, showing that $\rho_{\mathcal{K}}$ affects this measure. Both debiasing methods are able to attain flatter lines.

In Figure 5.1, we plot the differences of accuracies on predicting the target classes between the Truck and Zebra subsets. By definition, the ideal case is that this discrepancy should be near 0, so that Equality of Opportunity holds. We first take a look at the “bowling alley” class on the left side of Figure 5.1. When looking at the standard model (blue line), we see that it obtains lower values closer to $\rho_{\mathcal{K}} = 0.5$, and has sharp peaks as it gets closer to the boundary of 0.0 and 1.0. On the other hand, we see less drastic changes on our method as well as Adversarial Debiasing. The two are comparable across $\rho_{\mathcal{K}}$, with ours having less than or near to 5% difference whereas Adversarial Debiasing is slightly higher near $\rho_{\mathcal{K}} = \{0, 0.25\}$. Both also have lower discrepancies than standard training, with the exception of $\rho_{\mathcal{K}} = 0.75$. This seems to

be an outlier, and extra trials may be needed at that ratio. Nevertheless, this is promising in that our method does reduce the discrepancy and stays at a near constant amount across varying ratios.

On the right side of Figure 5.1, we plot the average discrepancies for \mathcal{K}^c . We quickly comment that all models are near 0 and comparable across ratios; this is expected (for standard training) and ideal (for the debiasing methods), because we keep these classes with a 50-50 split between pasted trucks and zebras.

5.1.2 Model Leakage

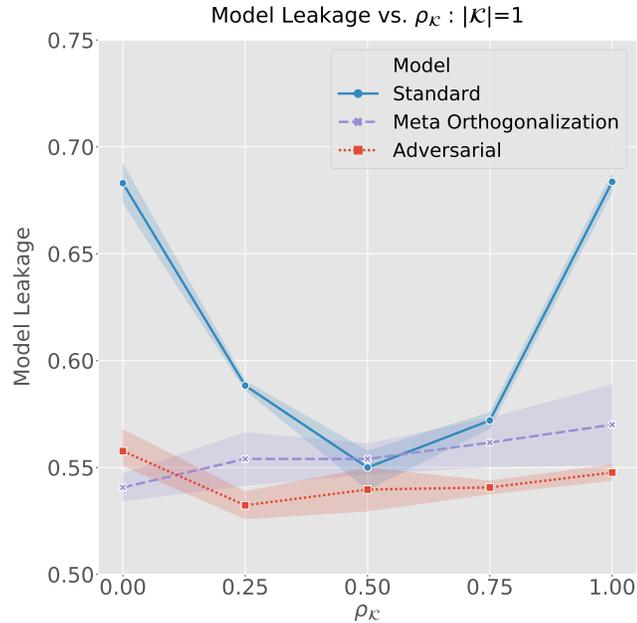


Figure 5.2: Comparing model leakage when modifying $\rho_{\mathcal{X}}$ in bowling alley. Parabolic trend in standard training, showing logit information increases as $\rho_{\mathcal{X}}$ deviates from 0.5. Both debiasing methods are able to flatten the curve.

Next, we view the differences in model leakage in Figure 5.2. This

measures how much predictive information is contained in the final logits of the respective networks; thus the ideal situation is similar to the previous fairness metric, since we want this to be as small as possible (50% or random accuracy) and as flat as possible (unaffected by $\rho_{\mathcal{X}}$). In the case of standard training, we see a similar parabolic trend, with its minimum at $\rho_{\mathcal{X}} = 0.5$. On the other hand, the two debiasing frameworks significantly decrease model leakage. In addition, although on average our method had higher model leakage, we remain statistically insignificant to the Adversarial Debiasing framework.

5.1.3 Projection Bias

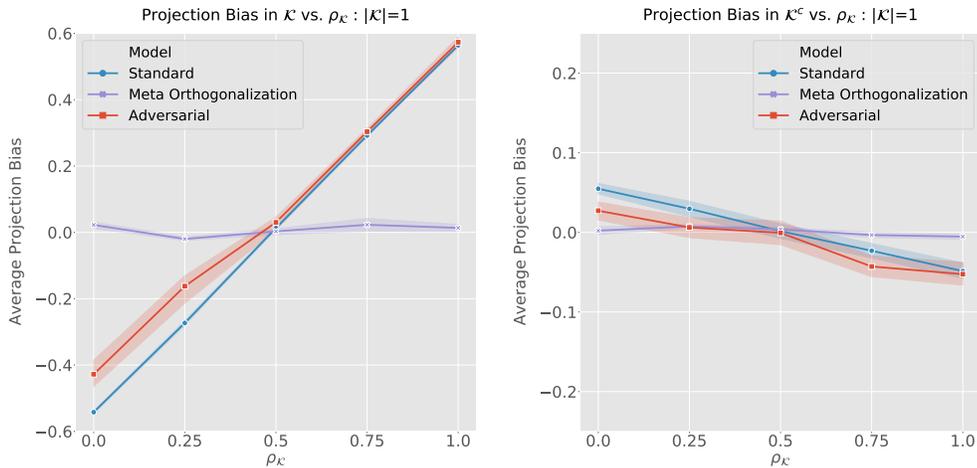


Figure 5.3: Comparing projection bias of models across varying $\rho_{\mathcal{X}}$. **Left:** Specific to the bowling alley class. **Right:** Averaged over every other class. Only our method results in near orthogonal embeddings. Adversarial debiasing is sometimes statistically closer to the 0 line than standard training.

We now view the projections of post-hoc image concept embeddings *w.r.t.* to the bias vector in Figure 5.3. On the left, we have the projection bias of the “bowling alley” concept, and clearly see that our method is nearly

horizontal at the 0.0 line, whereas Adversarial Debiasing obtains significantly higher correlation with $\rho_{\mathcal{X}}$. We expect this to happen with our method, as the regularization term is defined to do well on this metric, but we wanted to see how Adversarial Debiasing affects it as well. Specifically, Adversarial Debiasing can be significantly closer to 0.0 than standard training (such as at $\rho_{\mathcal{X}} = 0.0$, but not to the extent of our method).

On the right of Figure 5.3, we see that even in the unbiased set \mathcal{K}^c , we obtain near 0.0 average projection bias. Standard training and Adversarial Debiasing do not have as strong of a correlation, but can be seen to have a negative slope instead – this is mainly because in \mathcal{K}^c , these classes have more images with trucks and zebras when $\rho_{\mathcal{X}}$ is less than 0.5 and greater than 0.5, respectively.

5.1.4 Sensitivity Bias

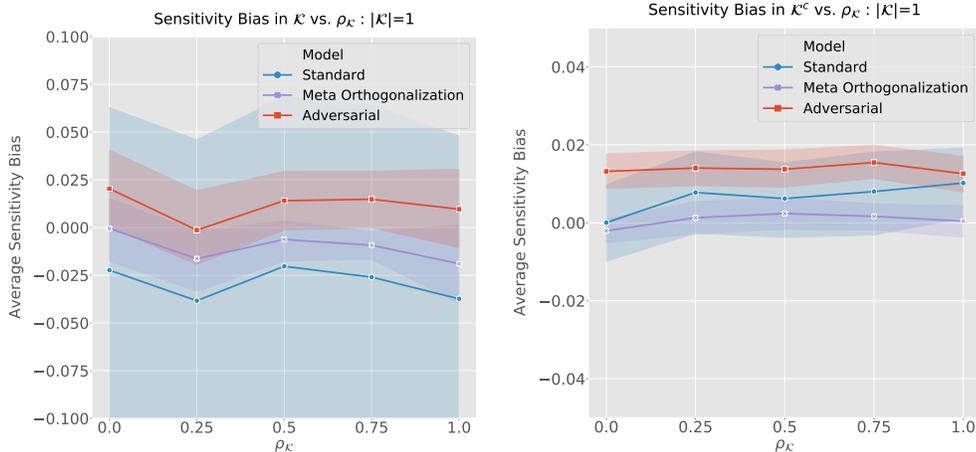


Figure 5.4: Comparing sensitivity bias of models across varying $\rho_{\mathcal{K}}$. **Left:** Specific to the bowling alley class. We see a higher variance for the biased bowling alley class in standard training. Both debiasing methods have comparable results. **Right:** Averaged over every other class. Our method is able to reach closer to 0 sensitivity bias and is statistically closer than Adversarial Debiasing.

Lastly, we look at the sensitivity biases of these models in Figure 5.4. This is similar to the previous metric, but instead of using a surrogate image concept embedding to represent a class, we directly use the class-specific gradient of the output logit in the models. The ideal case is that they are also nearly orthogonal, to indicate that the model’s confidence does not increase nor decrease if the intermediate representation moves in the bias direction (by Equation 3.10).

The main difference we note is the fact that the bowling alley class (left of Figure 5.4) tends to have a higher variance of sensitivity bias compared to the debiasing methods. We think this is because without any regularization

or constraints, the CNN can choose to bias towards any direction in the latent space. This may also explain why there is no clear linear trend *w.r.t.* $\rho_{\mathcal{X}}$. When studying the average sensitivity bias of the unbiased set \mathcal{K}^c (right of Figure 5.4), this variance is not as pronounced, since we have kept their ratios to be constant. We do note, however, that Meta Orthogonalization is statistically comparable in the left of the figure, but is clearly at the 0.0 line on the right. It is able to detect a slight bias in sensitivity and corrects it without an explicit regularization term.

5.2 BAM: Case Study II

We now extend the above analysis to $\mathcal{K} = \{3, 5, 7, 10\}$. The goal is to see if the same trends in the simpler Case Study I can hold when the biased set of classes \mathcal{K} grows. Because \mathcal{K} now contains multiple classes, we provide average metrics over the set, similar to how we showed this for \mathcal{K}^c in the previous study.

5.2.1 Equality of Opportunity

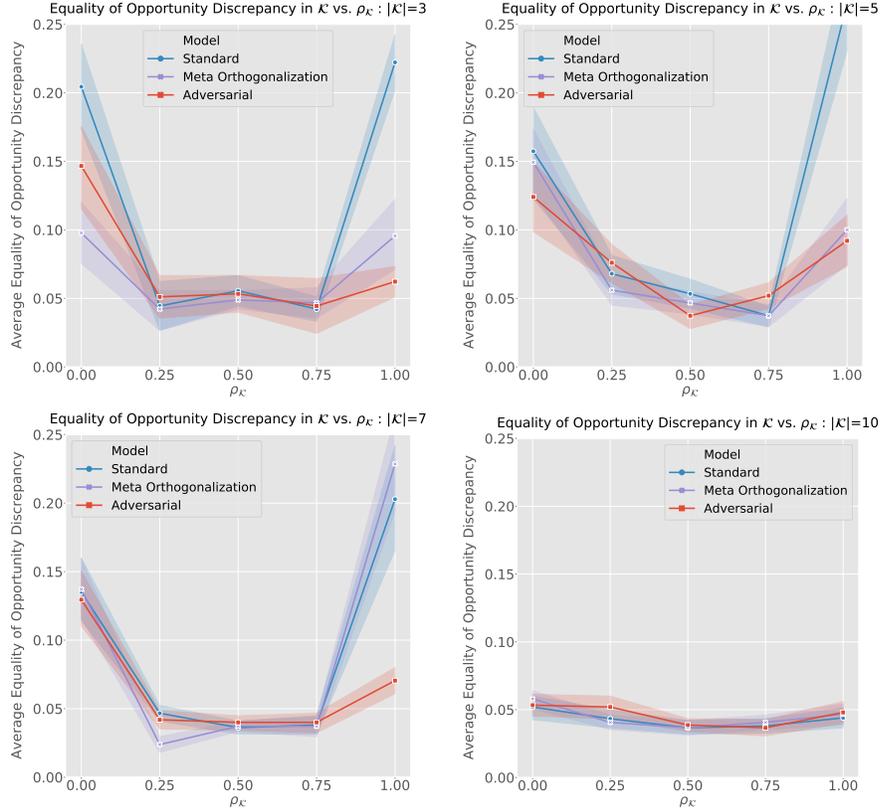


Figure 5.5: Comparing the opportunity discrepancy of models (in the **biased** set \mathcal{K}) of models across varying $\rho_{\mathcal{K}}$. **Top Left:** $|\mathcal{K}| = 3$. **Top Right:** $|\mathcal{K}| = 5$. **Bottom Left:** $|\mathcal{K}| = 7$. **Bottom Right:** $|\mathcal{K}| = 10$. The parabolic trend is consistent for standard training (blue lines), and our method reduces discrepancy statistically comparable to Adversarial Debiasing, but struggles at $|\mathcal{K}| = 7$ (bottom left).

Figure 5.5 contains the discrepancies in the **biased** set \mathcal{K} of classes, each denoting a specific size. Much like in Case Study I, we see the parabolic curve of standard training (blue lines). There is no clear connection with the extent of this parabola (i.e. the values at the endpoints), but it does not seem

to correlate with the size of the \mathcal{K} . In the cases of size 3 and 5, our method is statistically comparable with the Adversarial Debiasing method; however at $|\mathcal{K}| = 7$ case, we are near identical to standard training. An interesting thing to note is the fact that in the $|\mathcal{K}| = 10$ case, all models sustain a low and flat curve of opportunity discrepancy. The ratio does not affect these results, but the debiasing methods show no improvement. We think this is because every class in the dataset now has an equivalent distribution of trucks and zebras, and thus do not utilize this information to improve predictions.

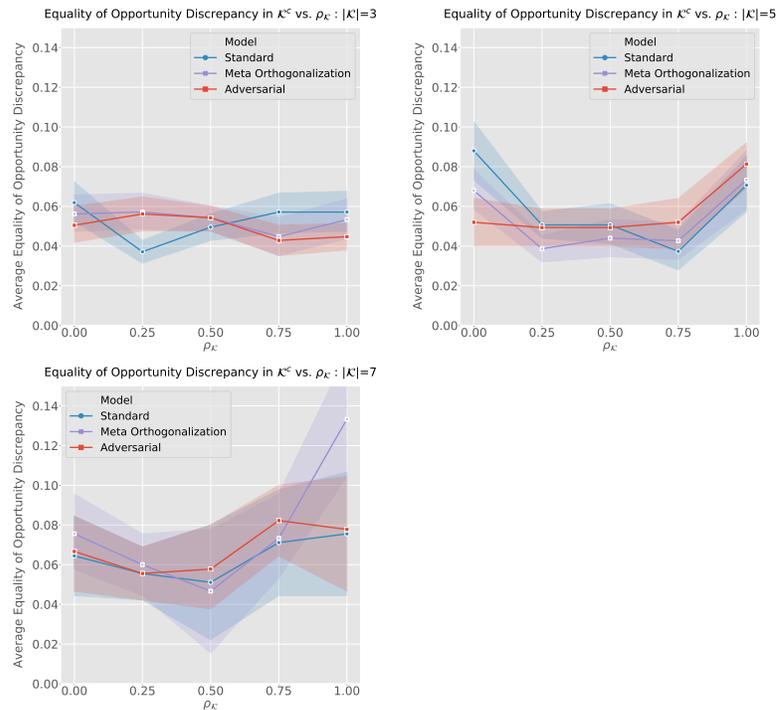


Figure 5.6: Comparing the opportunity discrepancy of models (in the **unbiased** set \mathcal{K}^c) of models across varying $\rho_{\mathcal{K}}$. **Top Left:** $|\mathcal{K}| = 3$. **Top Right:** $|\mathcal{K}| = 5$. **Bottom Left:** $|\mathcal{K}| = 7$. Although this is with classes with equal ratio, they begin to exhibit large discrepancies as $|\mathcal{K}|$ increases. Our method remains comparable except when $|\mathcal{K}| = 7$ (bottom left) at $\rho_{\mathcal{K}} = 1.0$.

In Figure 5.6, we plot the same discrepancies, but average in the unbiased set \mathcal{K}^c . We see that although these classes remain at a 50-50 ratio with trucks and zebras, as the biased set grows, the discrepancies begin to take parabolic shapes and even higher variance of results. Again, our method is comparable until $|\mathcal{K}| = 7$, but we do note that Adversarial Debiasing does not significantly decrease the discrepancy when compared to standard training. This is most likely because the bias is much less pronounced in these classes.

5.2.2 Model Leakage

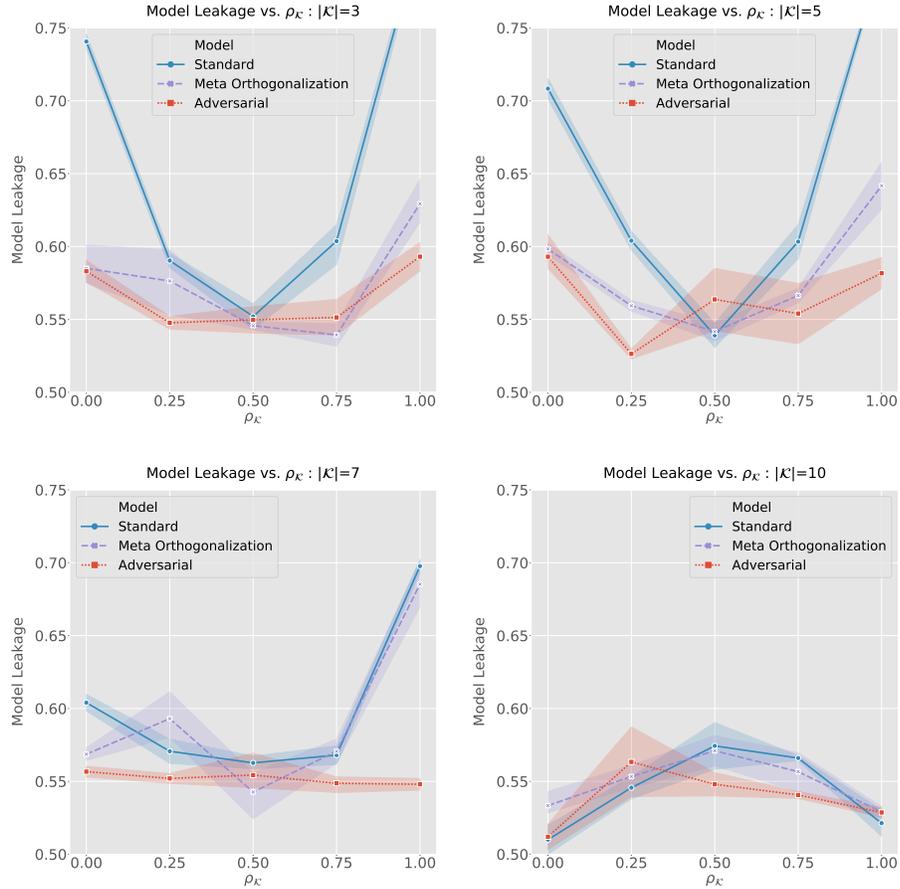


Figure 5.7: Model leakage vs. $\rho_{\mathcal{K}}$. **Top Left:** $|\mathcal{K}| = 3$. **Top Right:** $|\mathcal{K}| = 5$. **Bottom Left:** $|\mathcal{K}| = 7$. **Bottom Right:** $|\mathcal{K}| = 10$. We see less of a parabolic trend in standard training (blue line) as the biased set increases. As the size increases, our model begins to have more model leakage, whereas Adversarial Debiasing remains the same.

Figure 5.7 contains the comparisons of model leakage as we increase the size of our biased set \mathcal{K} . We see that for sizes 3 and 5, our method is again statistically insignificant to Adversarial Debiasing, and has significantly lower

leakage than standard training. However, once in the regime of 7 classes, Meta Orthogonalization fails at $\rho_{\mathcal{X}} = 1.0$, indicating that the protected attribute information has not been completely removed from the CNN’s logits. This could indicate that the regularization parameter may need to be accordingly adjusted for datasets with more bias, whereas the adversarial framework can have the same settings throughout.

Lastly, we see that in the trivial case of $|\mathcal{K}| = 10$, the model leakage is small and loses the upward parabolic shape for the standardly trained model. Interestingly, it is now parabolic downward, with the lowest leakage at the endpoints, i.e. containing 0 trucks or 0 zebras in the training set. Because this evaluation is done on the *balanced* test set, we see that it is probably the case that the network has learned to ignore the pasted images, i.e. use zero information from these patches. This is in line with what Yang and Kim [37] have found.

5.2.3 Projection Bias

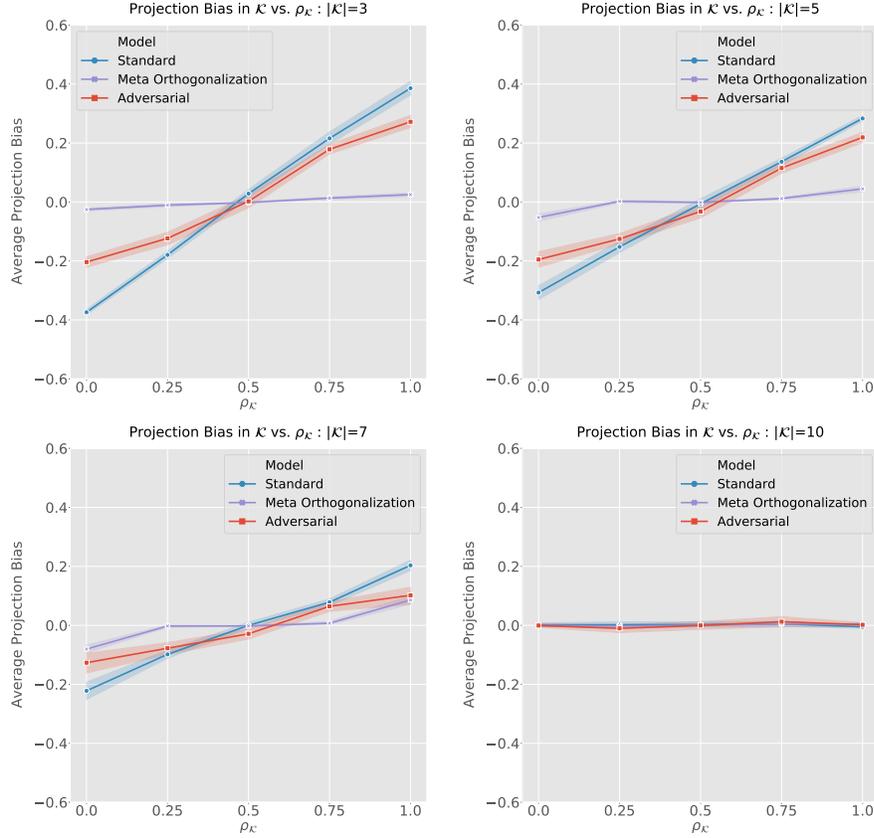


Figure 5.8: (in the **biased** set \mathcal{K}) of models across varying $\rho_{\mathcal{K}}$. **Top Left:** $|\mathcal{K}| = 3$. **Top Right:** $|\mathcal{K}| = 5$. **Bottom Left:** $|\mathcal{K}| = 7$. **Bottom Right:** $|\mathcal{K}| = 10$. Our method offers near 0.0 projection bias across all situations, but at $|\mathcal{K}| = 7$ (bottom left), we overlap with Adversarial Debiasing at extreme values of bias ratio.

Figure 5.8 now contains the resultant projection biases when adjusting the size of the biased set. The same phenomena as Case Study I is seen, as the blue curves always have a positive linear correlation with the ratio. However, we note that this correlations seems to decrease as $|\mathcal{K}|$ increases. The

goal remains the same though, in that these curves should be flat at the 0.0 line. Our method stays significantly closer to this zero line (than Adversarial Debiasing) until the case of $|\mathcal{K}| = 7$. Because this is the metric our method should perform near optimally in, this indicates that in this experiment, we could apply a stronger regularization term. In the experimental design, we kept a constant λ , to see how well a set value could generalize. Indeed, it seems that as the inherent bias of a dataset (denoted by the ratio as well as size of $|\mathcal{K}|$ increases, then we should apply a stronger penalty.

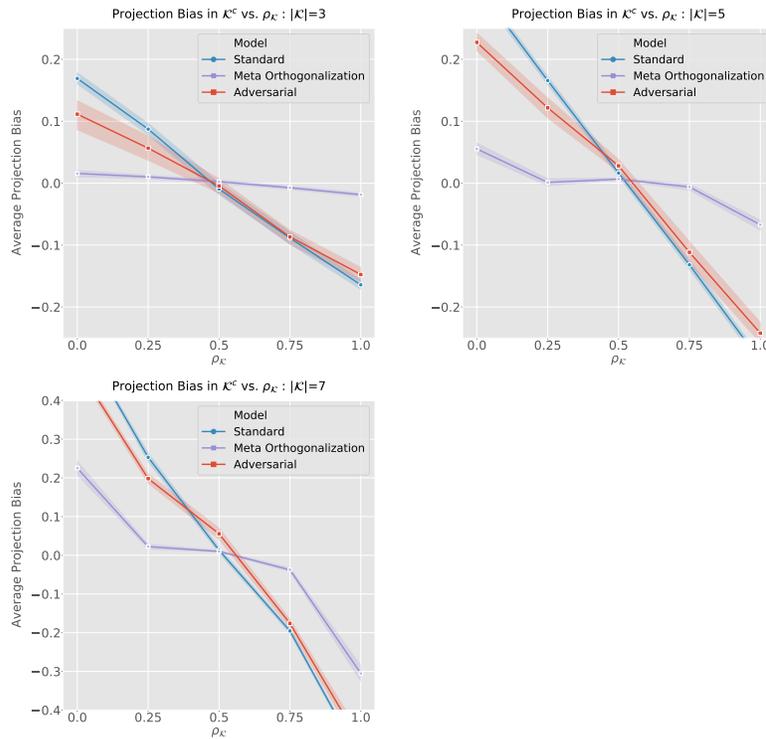


Figure 5.9: Comparing the projection bias (in the **unbiased** set \mathcal{K}^c) of models across varying $\rho_{\mathcal{K}}$. **Top Left:** $|\mathcal{K}| = 3$. **Top Right:** $|\mathcal{K}| = 5$. **Bottom Left:** $|\mathcal{K}| = 7$. We see our method is always closer to 0.0 projection bias, but notice that at a level of high bias $|\mathcal{K}| = 7$, these leftover classes end up being negatively correlated to the bias direction.

In Figure 5.9, we plot the average projection bias in the unbiased set \mathcal{K}^c . Unlike before, we can now see a strong negative correlation growing as the biased set size increases. This is likely due to the fact that \mathcal{K} begins to take majority in the dataset, and the images in the unbiased set \mathcal{K}^c are the only ones containing examples of either truck or zebras. This also supports our hypothesis that a stronger regularization term is needed, as these curves should be near flat, since our loss considers all classes, not just the biased ones.

5.2.4 Sensitivity Bias

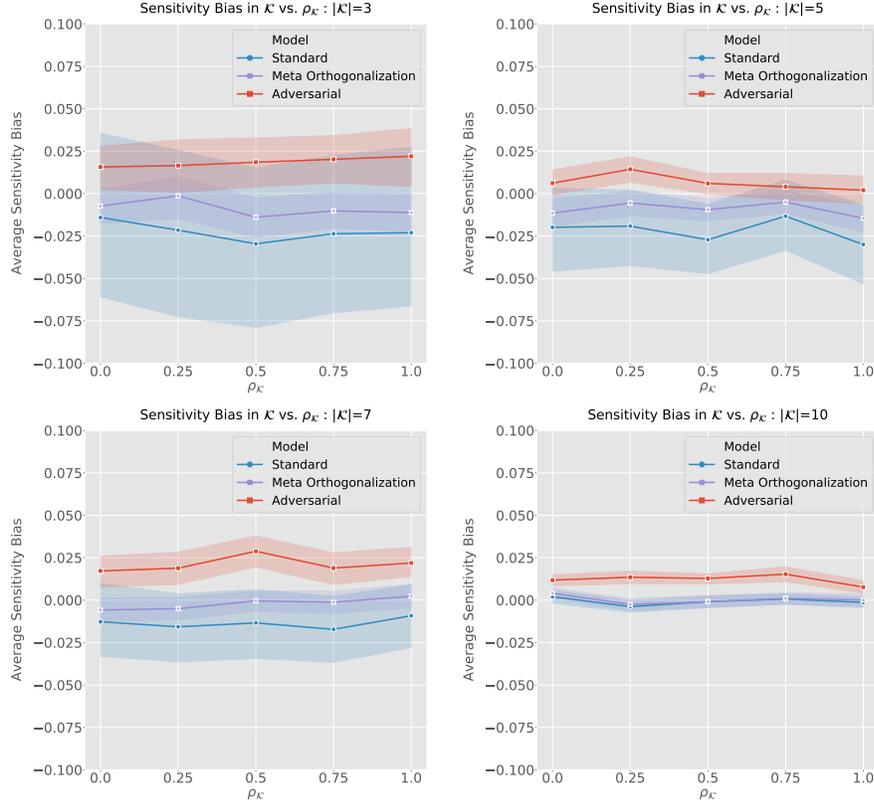


Figure 5.10: Comparing the sensitivity bias (in the **biased** set \mathcal{K}) of models across varying $\rho_{\mathcal{K}}$. **Top Left:** $|\mathcal{K}| = 3$. **Top Right:** $|\mathcal{K}| = 5$. **Bottom Left:** $|\mathcal{K}| = 7$. **Bottom Right:** $|\mathcal{K}| = 10$. Our approach is almost always significantly closer to 0.0 sensitivity bias across all ratios. Adversarial biases towards more positive values.

Although our method seems to perform well according to the previous metric, this does not directly translate to the behavior of the CNN in its downstream task. Thus we finally compare the sensitivity bias across these different situations. In Figure 5.10, we plot the sensitivity biases in the biased set, and note that near similar trends occur as in Case Study I. Namely, these

curves are all flat, and Adversarial Debiasing maintains a constantly small positive value, regardless of the changes in \mathcal{K} . Our method on the other hand, is significantly closer to the 0.0 line. Another thing to note is the fact that the variance in all the methods seems to decrease as the size of \mathcal{K} increases. We believe this is expected since the behavior of the model on the dataset becomes more and more similar as \mathcal{K} takes the majority of all images.

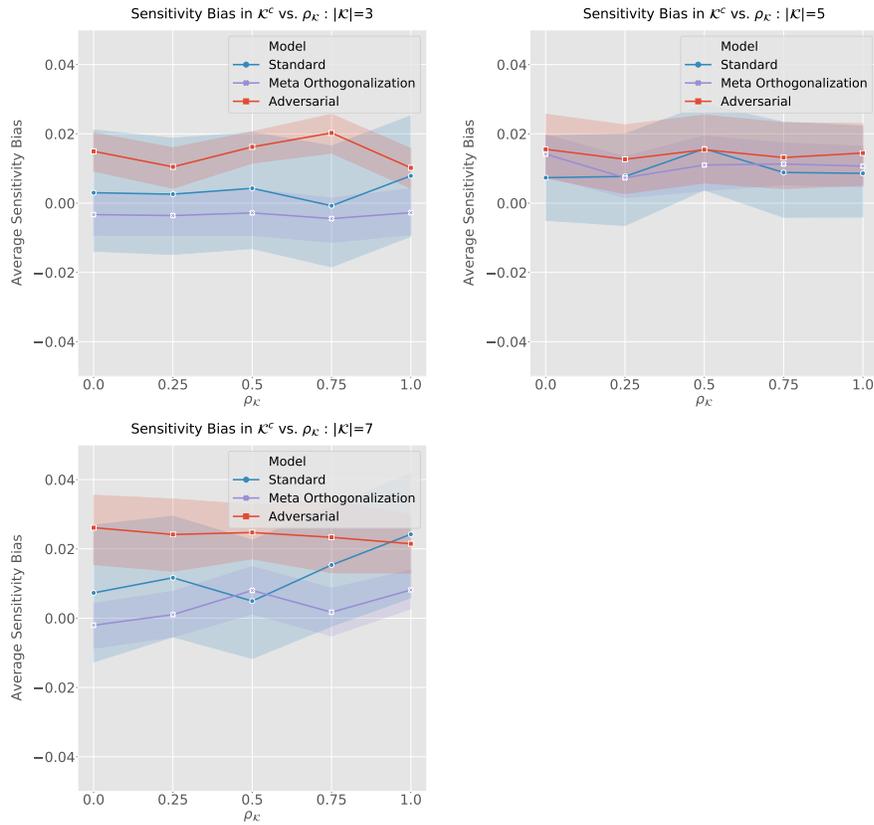


Figure 5.11: Comparing the sensitivity bias (in the **unbiased** set \mathcal{K}^c) of models across varying $\rho_{\mathcal{K}}$. **Top Left:** $|\mathcal{K}| = 3$. **Top Right:** $|\mathcal{K}| = 5$. **Bottom Left:** $|\mathcal{K}| = 7$. Our approach is almost always significantly closer to 0.0 sensitivity bias across all ratios. Adversarial biases towards more positive values.

Lastly, we observe almost the same trends in the unbiased set \mathcal{K}^c , except the trend in variance. This is also expected for similar reasons, since \mathcal{K}^c becomes more unique as the biased set takes over as the majority.

Chapter 6

Discussion

We now highlight a few key points we notice in our results. First, Meta Orthogonalization almost always performs comparably with Adversarial Debiasing across our evaluation metrics. The main situation that it fails is when $|\mathcal{K}| = 7$. We believe that applying a stronger λ would improve this case, due to our Projection Bias results. The main reason we keep a constant λ is to see the extent at which a single value could generalize, since this would give users more confidence when testing hyperparameters. This is important, in that real datasets may not have an expected measure of bias as we did when controlling $\rho_{\mathcal{K}}$.

Another good find is that in terms of the fairness metrics, Equality of Opportunity and Model Leakage, our method is able to obtain ideal results, even if we do not explicitly encode for this in the loss. Our regularization term does not enforce as strong of a constraint as an adversarial loss (i.e. removing necessary information for predicting the protected class), but can still be fair *w.r.t.* definitions in literature. In addition, our method is still able to retain predictive information for all concepts and protected attributes, but by applying the meta-loss, disentangles this information from the downstream classification task.

Lastly, through the varying parameters on \mathcal{K} , we provide more evidence for the hypothesis that more the more co-occurring sets of objects are, CNNs

tend to ignore this information. Specifically, when we increase the size of the biased set, the standard CNNs begins to bias towards \mathcal{K}^c , since they have more diverse examples. In addition, whenever \mathcal{K} encompasses all labels in the dataset, nearly zero bias across our evaluation metrics exists.

6.1 Future Work

Our setup naturally leads to the following ideas. First, orthogonality is not the only way to constrain the value of inner products. We can augment the debias loss to include a parameter $t \in [-1, 1]$:

$$\mathcal{L}_{\text{debias}}(\beta, t) = \sum_{i=0}^{N-1} \left(\frac{\beta_{c_i} \cdot \nu}{\|\beta_{c_i}\|_2 \|\nu\|_2} - t \right)^2. \quad (6.1)$$

This would allow for either weaker or stronger constraints on these concepts. For example, if a concept is strongly biased towards the positive protected attribute, one could use a $t > 0$ to enforce a weaker constraint, or $t < 0$ for a stronger constraint. We find in initial tests that the latter is difficult to train on, as this limits the predictiveness of the concept classifiers. It may be beneficial to be able to adaptively find a good t as well, since in real datasets, the bias may not be as uniform as we have in our experiments.

Second is to be able to apply this to an existing biased dataset. We find that it was difficult to understand the effect of our method without our setup in BAM. An initial hypothesis could be that certain protected attributes, such as gender, are not as easy to decorrelate compared to object co-occurrences. An example task that this could apply to is image captioning. Specifically, Hendricks et al. [18] tackle the case of vision models relying too heavily on gender cues to generate image captions, when gender is not the main subject

of topic. Our framework could be applied in this space, since it focuses on the convolutional portion of the model; Meta Orthogonalization has no dependence on the subsequent layers after the debiased latent space.

Lastly, the strength of our method also depends on the predictive power of the linear embeddings; for these more difficult tasks, it might be necessary to come up with a way to generate concept specific embeddings that are not restricted to a single linear layer. Some initial thoughts would be to use smaller fully connected layers as concept classifiers, but use some type of aggregated gradient vector as the final embedding. Further work would need to be done to show the arithmetic properties of this type of concept formulation; in addition, it may provide weaker signals for the CNN in Meta Orthogonalization since there would exist more layers between the embedding and the CNN's parameters. Nevertheless, we see this as a promising research direction for understanding and debiasing deep neural networks.

Chapter 7

Conclusion

In this work, we propose *Meta Orthogonalization* as a way to debias convolutional neural networks, inspired by previous work in NLP. We show that across various defined and newly proposed fairness metrics, our method performs comparably to Adversarial Debiasing, which has shown much success in previous literature. In addition, we provide a methodology to simulate various levels of bias and situations in a dataset, through the use of BAM [37], a recent dataset of object co-occurrences. Through extensive analysis on various trends of bias, we have shown promising empirical results for our method, indicating that there may be a strong connection to bias learned in CNNs and geometric properties of their latent spaces.

Bibliography

- [1] Machine learning glossary: Fairness. <https://developers.google.com/machine-learning/glossary/fairness>. Last Accessed: 2020-07-26.
- [2] Andrychowicz, M., M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. de Freitas. Learning to learn by gradient descent by gradient descent. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, page 3988–3996, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.
- [3] Angwin, J., J. Larson, S. Mattu, and L. Kirchner. Machine bias. *ProPublica*, 2016. URL <https://www.propublica.org/article/bias-in-criminal-risk-scores-is-mathematically-inevitable-researchers-say>.
- [4] Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140, 07 2015. doi: 10.1371/journal.pone.0130140. URL <http://dx.doi.org/10.1371/journal.pone.0130140>.
- [5] Beutel, A., J. Chen, Z. Zhao, and E. H. Chi. Data decisions and theoretical implications when adversarially learning fair representations. *CoRR*, abs/1707.00075, 2017. URL <http://arxiv.org/abs/1707.00075>.

- [6] Bolukbasi, T., K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In Lee, D. D., M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4349–4357. Curran Associates, Inc., 2016. URL <http://papers.nips.cc/paper/6228-man-is-to-computer-programmer-as-woman-is-to-homemaker-debiasing-word-embeddings.pdf>.
- [7] Calders, T., F. Kamiran, and M. Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18, 2009.
- [8] Chen, Z., Y. Bei, and C. Rudin. Concept whitening for interpretable image recognition, 2020.
- [9] Cleary, T. A. Test bias: Validity of the scholastic aptitude test for negro and white students in integrated colleges. *ETS Research Bulletin Series*, 1966(2):i–23, 1966. doi: 10.1002/j.2333-8504.1966.tb00529.x. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2333-8504.1966.tb00529.x>.
- [10] Dwork, C., M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311151. doi: 10.1145/2090236.2090255. URL <https://doi.org/10.1145/2090236.2090255>.
- [11] Finn, C., P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International*

Conference on Machine Learning - Volume 70, ICML'17, page 1126–1135. JMLR.org, 2017.

- [12] Fong, R. and A. Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [13] Ganin, Y., E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016.
- [14] Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Ghahramani, Z., M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>.
- [15] Guion, R. M. Employment tests and discriminatory hiring. *Industrial Relations: A Journal of Economy and Society*, 5(2):20–37, 1966. doi: 10.1111/j.1468-232X.1966.tb00449.x. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-232X.1966.tb00449.x>.
- [16] Hardt, M., E. Price, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In Lee, D. D., M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3315–3323. Curran Associates, Inc., 2016. URL <http://papers.nips.cc/paper/6374-equality-of-opportunity-in-supervised-learning.pdf>.

- [17] He, K., X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [18] Hendricks, L. A., K. Burns, K. Saenko, T. Darrell, and A. Rohrbach. Women also snowboard: Overcoming bias in captioning models. In Ferrari, V., M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 793–811, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01219-9.
- [19] Huang, L., L. Huang, D. Yang, B. Lang, and J. Deng. Decorrelated batch normalization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 791–800, 2018.
- [20] Hutchinson, B. and M. Mitchell. 50 years of test (un)fairness. *Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT* '19*, 2019. doi: 10.1145/3287560.3287600. URL <http://dx.doi.org/10.1145/3287560.3287600>.
- [21] Kamiran, F. and T. Calders. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*, pages 1–6, 2009.
- [22] Kaneko, M. and D. Bollegala. Gender-preserving debiasing for pre-trained word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1641–1650, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1160. URL <https://www.aclweb.org/anthology/P19-1160>.

- [23] Kim, B., M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. B. Viégas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2018.
- [24] Kim, M., O. Reingold, and G. Rothblum. Fairness through computationally-bounded awareness. In Bengio, S., H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4842–4852. Curran Associates, Inc., 2018. URL <http://papers.nips.cc/paper/7733-fairness-through-computationally-bounded-awareness.pdf>.
- [25] Lapuschkin, S., S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1), Mar 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-08987-4. URL <http://dx.doi.org/10.1038/s41467-019-08987-4>.
- [26] Lemoine, B., B. Zhang, and M. Mitchell, editors. *Mitigating Unwanted Biases with Adversarial Learning*, 2018. URL http://www.aies-conference.com/wp-content/papers/main/AIES_2018_paper_162.pdf.
- [27] Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In Fleet, D., T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.
- [28] Madras, D., E. Creager, T. Pitassi, and R. Zemel. Learning adversarially fair and transferable representations. In Dy, J. and A. Krause, editors,

- Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3384–3393, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL <http://proceedings.mlr.press/v80/madras18a.html>.
- [29] Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. URL <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf>.
- [30] Montavon, G., S. Bach, A. Binder, W. Samek, and K.-R. Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017. doi: 10.1016/j.patcog.2016.11.008. URL <http://dx.doi.org/10.1016/j.patcog.2016.11.008>.
- [31] Pennington, J., R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL <http://www.aclweb.org/anthology/D14-1162>.
- [32] Rebuffi, S.-A., R. C. Fong, X. Ji, and A. Vedaldi. There and back again: Revisiting backpropagation saliency methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [33] Simonyan, K., A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In

Proceedings of the International Conference on Learning Representations (ICLR), 2014.

- [34] THORNDIKE, R. L. Concepts of culture-fairness. *Journal of Educational Measurement*, 8(2):63–70, 1971. doi: 10.1111/j.1745-3984.1971.tb00907.x. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1745-3984.1971.tb00907.x>.
- [35] Wadsworth, C., F. Vera, and C. Piech. Achieving fairness through adversarial learning: an application to recidivism prediction. *CoRR*, abs/1807.00199, 2018. URL <http://arxiv.org/abs/1807.00199>.
- [36] Wang, T., J. Zhao, M. Yatskar, K.-W. Chang, and V. Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5309–5318, 2019.
- [37] Yang, M. and B. Kim. Benchmarking Attribution Methods with Relative Feature Importance. *CoRR*, abs/1907.09701, 2019.
- [38] Zhou, B., A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [39] Zliobaite, I. On the relation between accuracy and fairness in binary classification, 2015.