Analyzing Gender Biases in Visual Recognition Models

A

Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Master of Science

by

Jaspreet Ranjit

December 2021

APPROVAL SHEET

This

Thesis

is submitted in partial fulfillment of the requirements
for the degree of

Master of Science

Author: Jaspreet Ranjit

This Thesis has been read and approved by the examing committee:

Advisor: Vicente Ordóñez

Advisor:

Committee Member: Yangfeng Ji

Committee Member: Matthew Dwyer

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

December 2021

1 Abstract

With the rise of deep learning models which require exceedingly large amounts of data, there exists a need to examine the biases that are reflected in the applications of these models. For example, a visual recognition model can learn image representations of cooking that are closer to the representations of women than men, thus reinforcing a negative gender stereotype of women being homemakers. This thesis explores and analyzes these biases across state of the art visual recognition models. Deep learning models are reliant on large amounts of annotated data in order to be trained. Annotated data is difficult to collect and is often aggregated from human annotators or scraped from the Internet. As a result, these large, publicly available datasets can reflect societal biases. Labeled datasets require annotations provided by human labelers, which will reflect their individual biases. Furthermore, these biases can propagate into the model during training and potentially be amplified and reflected in the predictions. With rising concerns of discrimination and bias in deep learning, it is imperative to investigate the fairness and equity of these systems for all users.

Current bias identification pipelines target the explicit predictions of a model, often overlooking the implicit feature representations that contribute to biased predictions. The goal of this research is to investigate and compare gender biases across visual recognition models by quantifying bias relationships at the feature representation level. This is accomplished by exploring metrics that are able to capture the spatial relationships among classes in the feature representation of a deep neural network, and investigating factors that contribute to biases with respect to classes of images that co-occur with different genders. This work demonstrates that the source of this bias can be better understood by comparing the trend of feature representations for a group of classes across visual recognition models with different objectives. The work presented in this thesis serves as an exploratory step for a bias identification pipeline that explores gender bias relationships beyond the explicit predictions made by a model. This work can be extended to exploring other societal biases such as racial and religious biases. With the release of many deep learning models that have been trained on millions of images, we hope the work presented in this thesis aims at providing more transparency in how these models represent gender and encode bias at the feature level.

2 Gender Bias in Image Feature Representation

2.1 Definition of Problem

2.1.1 Overview and Motivation

While there have been many advancements in deep learning models in the fields of computer vision and natural language processing, the models are dependent on large amounts of available, annotated data. These models are known as foundational models [1]. Foundational models are neural networks that have been trained on a large amount of data and can be

reused for downstream tasks. There are a lot of models that the machine learning community has released where a large scale dataset is used to train a neural network that can be reused for various purposes. A big issue with these models is that they are not very accessible, the resources needed to train these models are highly concentrated in industry and even the data and the code required to reproduce their training is often not released. As a result, we're often required to use them as blackboxes. But they follow a general pipeline where, given some large scale dataset, they are trained using some neural network architecture on an objective such as image classification as illustrated in Figure 1.

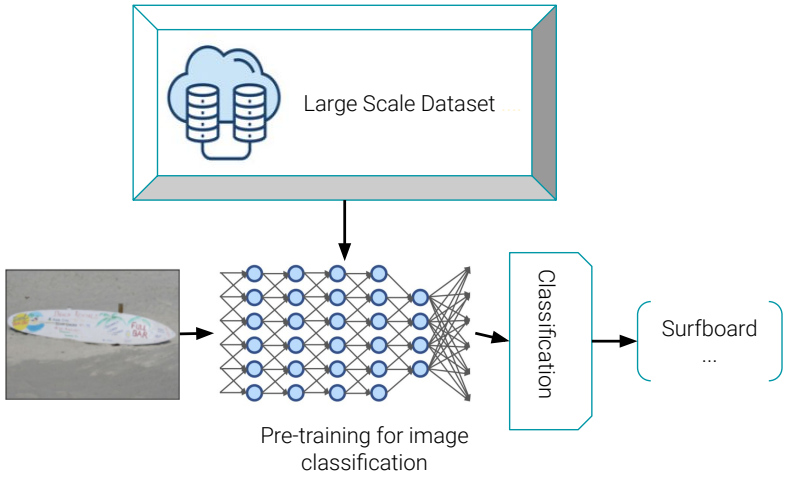

Figure 1: Foundational Models. Given some large scale dataset, train a neural network architecture on an objective such as image classification.

These foundational models are usually released by the machine learning community and have been pretrained on some large scale dataset. They can be used to train on a smaller dataset which is usually specific to the task you want to do. This process of retraining the model on top of the original pretrained model is called finetuning as illustrated in Figure 2.

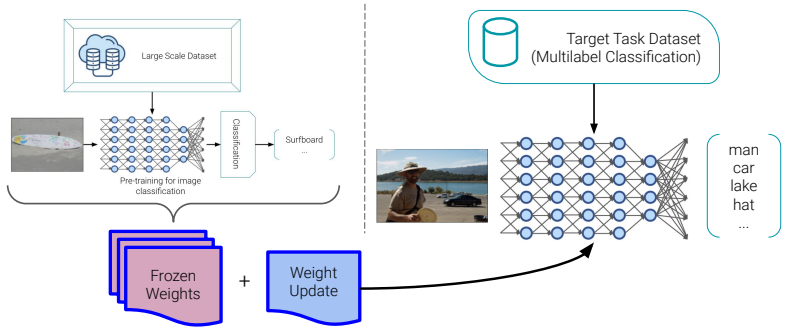

Figure 2: Finetuning Pipeline.

An example large scale dataset is the COCO (Common Objects in Context) [2], an object detection, segmentation and captioning dataset has over 330K images containing 80 object categories and serves as an important benchmark for many computer vision tasks [3, 4, 5]. ImageNet is another common benchmark with over 1000 object categories and over one million images [6]. Due to their ease of accessibility and scalability, many deep learning networks such as ResNet50 [3], SimCLR [7], MoCo [8] have been trained on these datasets and publicly released as pretrained models that can be finetuned on a downstream task. For example, ResNet50 has been pretrained on ImageNet-1k in a supervised setting. In order to adapt this model to a downstream task of object classification on the COCO dataset, the pretrained ResNet50 model can be finetuned on the COCO dataset.

Datasets such as ImageNet and COCO provide a universal benchmark for research studies, are rich in annotations and metadata, and provide a diverse set of classes for networks to learn. However, these datasets are often aggregated from the Internet and as a result can reflect harmful biases present in our society. Furthermore, the annotations are collected and verified by humans, making them prone to each individual's own implicit biases. For example, in the imSitu [9] training set, 33% of cooking images have man in the agent role while the rest have woman [10]. Bias towards women and cooking can be learned and amplified by models where a trained Conditional Random Field reduced the number of men labeled as cooking to 16% [10]. Furthermore, [11] found that the COCO dataset is skewed towards lighter skinned people over darker skinned people and males over females; images with lighter skinned people are 7.5x more common and images with males are 2.0x more common than images with females. These biases have been shown to propagate into the network during training and adversely affect the fairness and equity of the predictions. For example, models are at risk of amplifying biases that exist in the dataset by compromising protected attributes such as gender and implicitly making associations that reinforce negative gender stereotypes. In more critical applications, it is important to ensure the quality of the predictions are fair and equitable for all user groups, and as a result, there is a need for a way to characterize these biases and compare them across models.

Bias and fairness in machine learning have been addressed previously in studies such as [11] and [10], but these studies primarily dissect the explicit predictions made by the model. This thesis explores and quantifies how these biases are reflected in the feature representation of visual recognition models. Not only does this approach explore a finer level of granularity to explore these bias relationships, but it also provides more transparency and insight into how the model is learning these relationships. In addition, we also explore how transfer learning impacts these biases and analyze several variables to better discern the source of bias. We focus on visual recognition models and we finetune these models on the task of multilabel image classification to examine biases in the feature representation space. More specifically, we curate a dataset of images for bias analysis, and identify metrics to analyze the intra-cluster variation for a given class. The representation of intra-cluster variation serves as a proxy to identify how biases for different classes are represented in the feature space.

We can compare representations of classes of a pretrained and finetuned model to understand how biases are represented before and after transfer learning. Furthermore, we can compare the feature representations across models to understand how the architecture of a network, the dataset it was pretrained on, and the setting it was pretrained with impacts the biases in the feature space.

2.1.2 Technical Challenges

This thesis addresses several key technical challenges in studying bias in neural networks. When investigating biases in neural networks, there exists a need for a dataset that is labeled with the biases in question. However, this data is extremely rare and bias datasets that are labeled by humans are still prone to error and prejudice in human judgement. This makes it difficult to evaluate how bias shows up in a dataset and a network. For example, if we are trying to study gender bias in object recognition tasks, it is necessary to have a dataset of labeled images with gender. Publicly available datasets such as COCO [2] do not contain such labels making it difficult to evaluate these biases in a standardized fashion.

Furthermore, examining and quantifying biases at the feature representation level for comparison across models is particularly difficult. It is challenging to quantitatively compare the hidden layer representations of neural networks because the features are distributed across a large number of neurons [12]. Because of differences in the latent space representation of the features, the feature representations of two models cannot be compared directly, necessitating the use of more qualitative methods to identify biases.

2.2 Definition of Bias

2.2.1 Intra-Class Variation

For a given model, we represent the biases of a model with respect to a given class of images by evaluating the intra-class and inter-class variation at the feature representation level. Given an analysis set D with n classes where each class of images has k images, a similarity/distance metric $f(x,y)$, and a model, $M(*)$, we evaluate this model on each class $n \in D$, to generate a set of features: $F \in \mathbb{R}^{n \times k \times d}$ where d is the feature embedding size for model $M(*)$. These features are extracted from the penultimate layer of the model $M(*)$. We then use $f(x,y)$ to evaluate the intra-class variation of a single class $i \in n$, by taking two random samples $x, y \in F[i,:,:]$ of the feature data corresponding to class $i \in n$ and calculating the distance $f(x,y)$ between the two samples. We repeat this random sampling process for T iterations for each class $i \in n$, resulting in T values representing the intra-class variation for a class of images. We also compute a one-sample t-significance test, and average these T values to get μ_i for a single class of images. In addition, we also compute $\min(T)$ and $\max(T)$ to represent the range of variation across all the classes. We calculate one such μ_i value for each class $i \in n$ thus providing us with a comparison of the intra-class variation in an analysis set D, for a given model M using a metric $f(x,y)$. The intra-class variations

$\{\mu_i\}_{i=0}^n$ serve as a proxy to identify biases of a model and we can use $\{\mu_i\}_{i=0}^n$ to compare relative biases across models. The variations provide an intuitive representation of how well a model clusters a class of images in the latent space. This clustering can be compared *relatively* to other classes. Ideally, objects should not be biased with respect to gender and we can examine these biases by evaluating similarity/distance in the latent space. For example, we can examine the embeddings of images of men, and women with respect to computer, and the distance between man and computer and woman and computer should be equidistant. Our definition of bias can capture this phenomenon by comparing the intra-class variation across classes which are suspected to contain biases. This idea of bias was introduced in [13] where biases were examined in word embeddings and we extend this work to analyze image embeddings. This pipeline is illustrated in Figure 3

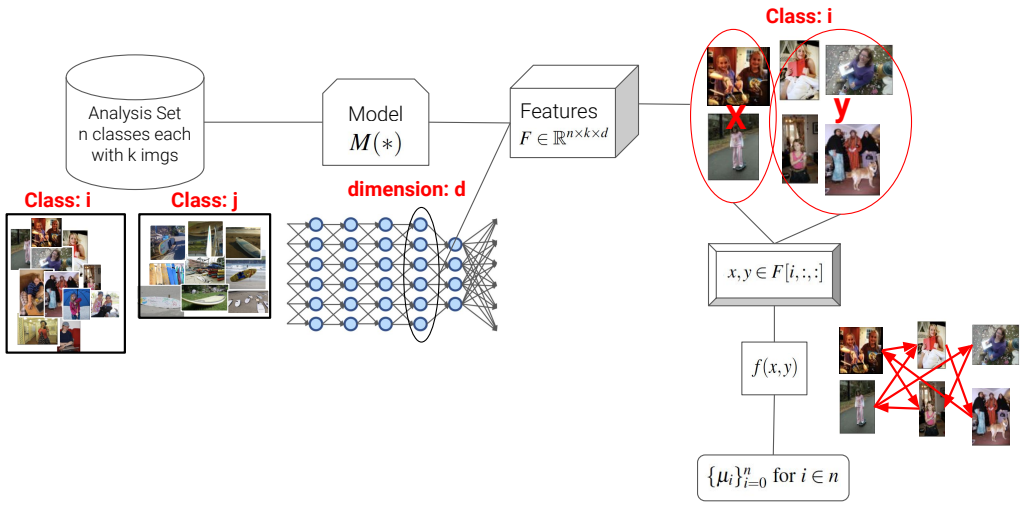

Figure 3: Intra-class metric calculation. Given an analysis set, with n classes each with k number of images, and a pretrained model, or a pretrained model that has been finetuned on some downstream task. We use the classes in our analysis set to extract a set of features from the hidden layers of the network. These features are the representation of the classes of images in some hyperdimensional space. To examine biases at the class level, we take a single class and generate two random folds of this data. We grab their respective features from the features we generated from the model. We then use some similarity metric, f(x,y) to calculate the similarity between the two folds of the features corresponding to the images in the class we are examining and we can repeat this process for a set number of iterations to get an average similarity score for a given class. Intuitively, this μ score for each class shows how well the model represented a class where a higher mu score implies that the model represented a specific class better.

2.2.2 Inter-Class Distance

Alternatively, we also examine biases between two classes $i, j \in n$ where n is the number of classes in an analysis set D and each class n has k number of images. We take two random samples $x \in F[i,:,:]$ and $y \in F[j,:,:]$ and calculate $f(x,y)$. We repeat this random sampling for T iterations to get T values representing the distance between two classes $i, j \in n$ in the latent space. We take the average of these T values to get $\mu_{i,j}$ which represents the average distance between two classes. The $\mu_{i,j}$ can be calculated for all pairs of classes in the analysis set D for a specific model $M(*)$ and a metric $f(x,y)$. This pipeline is illustrated in Figure 4

To examine biases between classes, we use a very similar pipeline but instead, we take two random folds of data from two different classes, and we calculate the similarity between these folds of data to get a mu score that represents the similarity between two different classes. This approach allows us to more directly compare the similarity between two different classes and we can calculate this score for all pairs of classes in our analysis set.

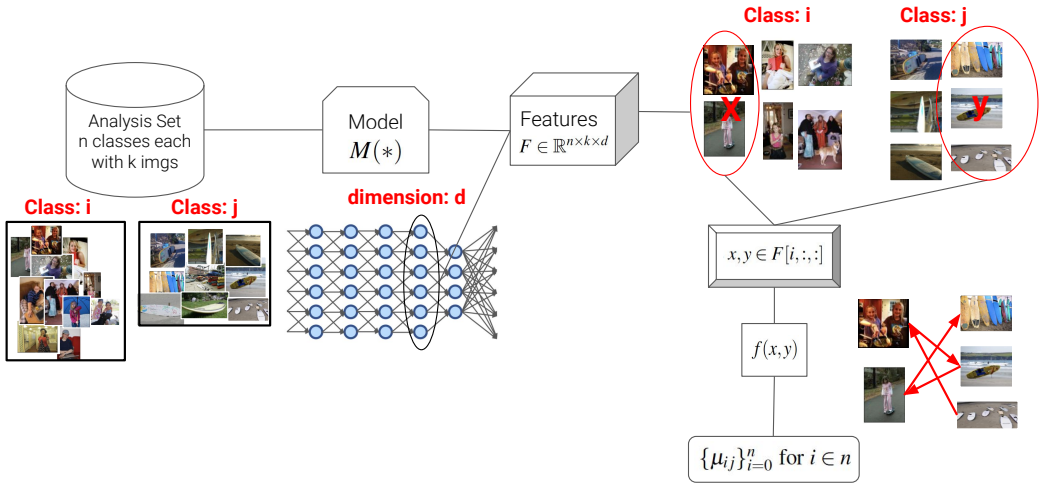

Figure 4: Inter-class metric calculation. To examine biases between classes, we use a very similar pipeline to the intra-class calculation, but instead, we take two random folds of data from two different classes, and we calculate the similarity between these folds of data to get a mu score that represents the similarity between two different classes. This approach allows us to more directly compare the similarity between two different classes and we can calculate this score for all pairs of classes in our analysis set.

2.3 Metric Interpretation

The intra-class variation and the interclass distance both use a metric $f(x,y)$ that calculates the similarity between two sets of features in the latent space. We chose to test standard

pairwise cosine similarity to represent how well a model clusters a class of images and to test the similarity between two classes of images. We chose cosine similarity as a starting point to analyzing biases in the latent space for images because it is relatively easy to interpret. We want to use the intra-class variation and inter-class distance to interpret the biases of a model and compare biases across models. However, since each model's features are represented in a different latent space, we can instead compare the relative values of this metric across models. For example, given models $A(*)$ and $B(*)$, and an analysis set D with n classes, we can compute $A_{avg} = \{\mu_i\}_{i=0}^{n}$ and $B_{avg} = \{\mu_i\}_{i=0}^{n}$ where A_{avg} and B_{avg} represent the intra-class variation for n classes. We can compare the Spearman's coefficient [14] between A_{avg} and B_{avg} which shows us if two variables are monotonically related even if their relationship is not linear. This can also be computed for the inter-class distance between two different classes. Qualitatively, this gives us a way to compare two models in terms of their bias relationships where a higher correlation implies the biases of the two models are similar. We use our definition of bias to test how different variables contribute to bias.

2.3.1 Hypothesis

1. Given evidence of explicit biases in the predictions of models, we hypothesize that there exist biases at the feature representation level of a visual recognition model and we can identify and reasonably quantify these biases using standard metrics such as cosine similarity.

2. We can identify which factors contribute most to biases at the feature representation level by evaluating the change in bias with respect to the following variables for a given model: which dataset the model was pretrained on, the training setting (supervised vs. self-supervised), network architecture, and the dataset the model was finetuned on. We speculate that each of these variables impacts the biases of a model in the feature representation space in a different way.

3. We can qualitatively compare biases across models using our definition of bias and discern which factors contribute most to a model's representation of bias in the feature space.

2.3.2 Contributions

The primary goal of this thesis is to investigate and quantify biases at the feature representation level, compare these biases across visual recognition models, and identify how characteristics of a model contribute to these biases. Deep neural networks are high in complexity and often are treated as a black box in many industrial applications. As a result, they do not typically undergo standardized software testing and can be prone to learning harmful biases. This lack of transparency can further lead to discrimination and inequities in our technology that impact end users. As a result, this thesis focuses primarily on quantifying

these biases in the feature representation space to gauge a better understanding of how models encode these biases. This serves as a vital first step in better understanding how biases propagate through a network and how transfer learning further affects these biases. The contributions of this work are as follows:

1. Aggregation of a labeled dataset for measuring biases across classes of images that may potentially exhibit gender bias. Separate *analysis datasets* are aggregated for both COCO2017 [2] and Open Images [15].

2. Identification of metrics and proposal of a methodology for identifying biases in hidden layers of a model. Given an analysis set of images, these metrics provide a quantitative measure to capture the intra-cluster variation of a class of images that can be compared relative to other classes.

3. Development of a qualitative methodology to compare several different models for biases using our improved metrics for bias on the aggregated analysis dataset. These models have been pretrained on different datasets such as ImageNet-1K, ImageNet-20k and COCO with a supervised or self-supervised setting. We finetune each of these models on the COCO and Open Images dataset. Using the intra-class variation metric and inter-class distance metric, biases are compared within a model before and after finetuning. Biases are also compared across models, along with a qualitative exploration of the factors which contribute most to a model's bias.

2.4 Related Work

2.4.1 Profiling Biases in Datasets

Previous studies have examined biases in datasets by studying the distribution of objects and categories in the annotations. [16] profiles the COCO 2014 [2] dataset for racial biases in image captions and studies bias propagation pathways in neural networks. Furthermore, [17] evaluates popular recognition datasets for relative data bias and provides insights into how biased datasets that serve as popular benchmarks for deep learning models can impact the notion of accuracy and effectiveness of a model's prediction. Beyond analyzing the dataset for biases, [18] addresses the issue of dataset bias by learning shared parameters across datasets which serves as an approximation to an unbiased dataset. We have focused on understanding how biases present in these benchmark datasets can propagate through the network and are reflected in the feature representation space.

2.4.2 Biases in Natural Language Processing

In addition to computer vision, biases have also been studied in natural language processing. [13] proposes a methodology to debias word embeddings by removing gender stereotype associations while retaining neutral indicators of gender. Their proposed methodology

identifies the gender subspace that represents the biased embedding space, and then equalizes the distance between words to reduce biased associations. They identify metrics to quantify biases in these word embeddings and show that the resulting debiased word embeddings can be used in downstream tasks without amplifying biases. Furthermore, [19] quantifies and mitigates gender biases in ELMo's contextualized word vectors by analyzing how ELMo encodes gender, and examining the corpus used for training ELMo. They found that this corpus has a skew towards male entities and this bias propagates into downstream applications. They propose a training time data augmentation technique and a test time embedding neutralization technique to mitigate these biases. We apply principles from bias identification in the word embedding space to the feature representation space in visual recognition models.

2.4.3 Bias Measurement and Identification

Our work is primarily focused on measuring and identifying biases across pretrained models for visual recognition. [20] introduces DeepInspect that measures biases in image classifiers by testing for class property violations. This work targets biases at the class level and proposes a testing technique to identify class based bias errors in deep neural networks. [21] is another approach that focuses on how the model represents its input data. InsideBias relies on the analysis of the learned features for a group of images and can detect biases of a model with a very small subset of images. This framework is primarily restricted to biases in facial recognition systems. [22] analyzes racial, gender and intersectional biases in state of the art unsupervised models. They adapt bias tests such as iEAT and embedding association tests designed for contextualized word embeddings and apply these methods to the image domain and evaluate how well these tests are able to identify biases in an unsupervised setting. We propose a bias measurement framework motivated by efforts in [13, 20, 21, 22] that explore biases at the feature representation level in the image domain. We extend this work by proposing metrics that capture the intra-cluster variation of image classes to characterize a model's internal feature representation for a set of categories. We also investigate how these representations change with transfer learning and perform a comparative study across visual recognition models to qualitatively analyze how network architecture, training methodologies, and datasets impact biases.

2.4.4 Debiasing Approaches

More recently, bias identification and debiasing pipelines have focused on identifying how models can amplify existing biases in datasets and how these biases are reflected in the predictions of a model. [10] finds that models trained on biased datasets further amplify these biases and they propose a framework that introduces corpus level constraints that balances the co-occurance of gender with other objects in the prediction task to reduce bias amplification in the predictions. In addition, [23] proposes a method that surpasses the need for human

labeled bias datasets that could be prone to human reporting bias to identify unknown biased attributes. They frame the bias identification task as an optimization problem in a generative model's latent space. They propose a network that is able to identify biased attributes of a class of images by optimizing a loss function dependent on a fairness criteria.

2.5 Methodology and Setup

2.5.1 Variables Impacting Bias

1. Models are pretrained on large datasets such as ImageNet [6] and COCO [2], and the weights are released for finetuning. We hypothesize that different pretraining datasets contribute to biases in the feature representation space. We test the following different pretraining datasets:
   (a) ImageNet1K
   (b) ImageNet 21K
   (c) 400M Images collected from the web [24]
2. An open problem in bias exploration is the impact of the network architecture on biases. We hypothesize that the network architecture has an impact on how the biases are represented in the feature space. We test the following network architectures:
   (a) ResNet18 [4]
   (b) ResNet50 [24]
   (c) CLIP ViT/32-B [24]
3. Furthermore, models are trained in different settings. We hypothesize that different training settings could impact the model's biases. Supervised learning uses labeled training data whereas self-supervised learning learns from unlabeled training data. We consider the following models in a supervised vs. self-supervised setting:
   (a) Supervised: ResNet18, ResNet50, BigTransfer ResNet50 [25], CLIP: ViT/B-32
   (b) Self-Supervised: Moco ResNet50
4. We hypothesize that finetuning impacts the biases of a model and the dataset used to finetune the model could introduce new biases into the feature representation space. We finetune the models on two datasets: COCO2017 and Open Images v4 and examine the change in biases. In summary, we test the following models:
   (a) ResNet18: A residual learning framework that solves the vanishing gradient problem by allowing gradients to flow through the skip connections from deeper layers to initial filters. This model has 18 layers. [4]
   (b) ResNet50: Same idea as ResNet18, but with 50 layers. [4]

(c) CLIP: ViT/B32: The CLIP model learns visual concepts from natural language supervision and performs zero-shot object classification tasks. We test the Vision Transformer: ViT/B-32 from the CLIP model. [24]

(d) MoCo ResNet50: Momentum Contrast for Unsupervised Visual Representation Learning, MoCo reframes the task of contrastive learning as a dictionary lookup. [8]

(e) BigTransfer ResNet50: Leverages the potential of pretraining on large scale datasets to pretrain ResNet50 on ImageNet21K for downstream tasks [25]

2.5.2 Bias Analysis Sets

As detailed in Technical Challenges, one of the primary challenges in analyzing deep neural networks for biases is the need for a labeled dataset that includes annotations regarding the bias in question. Furthermore, previous works tend to examine biases of individual images whereas we examine biases at the class level. We address this need for a dataset by collecting a subset of images, called an analysis set, from the benchmark COCO 2017 dataset [2] and the Open Images dataset [15]. In this work, we primarily want to examine gender bias and thus we choose categories that allow us to compare the co-occurence of genders with different objects. For example, if we want to examine the biases of man and woman with respect to an object such as a car, our dataset would need to include at least five classes: [man, woman, car, man+car, woman+car, random] where the gender+object (man+car, woman+car) categories include images with the gender and the object and ideally no other objects. We also include a random category that has a random subset of images from the dataset that serves as a baseline for comparison with our categories of interest. To generalize our methodology for collecting an analysis set, we come up with a list of objects we are interested in, and develop a dataset that includes the following categories for each object: [man, woman, object, man+object, woman+object, random]. As a result, if we are interested in examining biases of three objects with respect to gender, the baseline analysis set would include 12 classes. For the purposes of this study, we collect analysis sets from the COCO and Open Images dataset. The collection procedures and details of these analysis sets are explained in the following sections.

COCO 2017 Analysis Set. The COCO 2017 dataset has over 200K labeled images with 80 object categories. COCO 2017 serves as a benchmark metric for many object recognition tasks in computer vision and as a result, it serves a good starting point for testing our bias measurement framework. Furthermore, it has been shown to reflect negative gender stereotypes by exploratory data analysis studies so we wanted to examine the representation of these images in the feature space. We used the object annotations for each of the images to extract the analysis set. We wanted to examine gender with respect to the following object categories: [car, refrigerator, surfboard]. Because COCO does not have explicit 'man' and 'woman' annotations in the images, we first segmented the dataset by the category 'person' and the object categories, and manually sifted through this segmented dataset to develop an

analysis set with the categories described in Table 1 along with the count of images in each of the categories. The COCO2017 Analysis Set section in the Appendix A details examples from each class. We posit that because we manually sifted through the images to extract the analysis set, it was not necessary to have many images in each of the classes as there is very little noise. For example, we ensured that the man+object categories did not contain any woman labels so we could more accurately preserve the class identity.

Class	Number of Examples
man	12
woman	15
random	20
stopsign	44
car	12
car+man	9
car+woman	6
refrigerator	18
refrigerator+man	8
refrigerator+woman	9
surfboard	14
surfboard+man	23
surfboard+woman	16

Table 1: COCO2017 Analysis Set: Categories and examples per category

Open Images Analysis Set. The Open Images Analysis dataset is a much larger dataset than COCO 2017 and includes over 9 million varied images with rich annotations. The images contain 8.4 annotations on average and thus serves as a good counterpart to the COCO dataset for analysis. The Open Images analysis set was collected in a more automated way. We wanted to examine the following object categories with respect to gender: [car, sport equipment, fashion accessory, mammal]. These categories were chosen based on their class frequency in the original dataset. We relied solely on the semantic labels of these images to aggregate the analysis set ensuring that each class in the analysis set did not contain overlapping images with other classes. Table 2 provides an overview of this analysis set. It's clear that this dataset contains a lot more noise, and as a result, we collected more images per class. Section Open Images v4 Analysis Set in the Appendix A contains examples from each of these classes.

Class	Number of Examples
man	150
woman	150
random	150
stopsign	22
car	150
car+man	150
car+woman	49
sports	150
sports+man	120
sports+woman	32
fashion	150
fashion+man	52
fashion+woman	150
mammal	150
mammal+man	150
mammal+woman	150

Table 2: Openimages v4 Analysis Set: Classes and examples per Class

2.5.3 Experimental Setup

We want to test each how of the defined variables in the Section- Variables Impacting Bias: pretrained dataset, network architecture, pretraining setting and transfer learning, impacts how biases are represented in the feature space for a given model. The feature space is often represented by non-linear associations and as a result, is very difficult to interpret intuitively. As a first step in measuring biases in the feature representation space, we wanted to analyze the effectiveness of using standard, easily interpretable metrics to capture the representation for a given class. As a result, we use our definition of bias in Section- Definition of Problem with cosine similarity to represent the similarity between two sets of features. For each given model $M \in$ {ResNet18, ResNet50, CLIP:ViT/B-32, MoCo ResNet50, and Big Transfer ResNet50}, we finetune the model on COCO2017 and Open Images v4 for multilabel image classification to a comparable mean average precision and F1-Score. The specific metrics for each model are detailed in Tables 3 and 4 for finetuning on COCO2017 and Open Images v4 respectively. For an analysis set D, with n classes where each class has k number of images, and the model M has a hidden feature dimension: d, we extract two sets of features: pretrained-features, finetuned-features, each of size $n \times k \times d$. Each model is only evaluated on the analysis set corresponding to the dataset the model was finetuned on. For example, if a model M is finetuned on COCO2017, it is only evaluated on the COCO2017 analysis

set. We then use our definition of bias to calculate the intra-class variation for each set of features, and the embedding distance between pairs of classes.

Model	Pretraining Dataset	Pretraining Setting	Epochs	Learning Rate	Optimizer	mAP	Micro F1
BiT-M-R50x1	ImageNet-21k (20M)	Supervised	15	0.003	SGD, m: 0.9	0.7527	0.827
ResNet50	ImageNet-1k (1M)	Supervised	15	0.001	SGD, m: 0.9	0.7024	0.8363
ResNet18	ImageNet-1k (1M)	Supervised	40	0.1: reduce on plateau	SGD, m: 0.9, wd:1e-5	0.7443	0.7307
CLIP: ViT-B/32	400M images from web	Supervised	20	0.001	SGD, m: 0.9	0.7053	0.7929
MoCo ResNet50	ImageNet-1k (1M)	Self Supervised	20	0.1: reduce on plateau	SGD, m: 0.9, wd:1e-5	0.6268	0.6460

Table 3: Results from Finetuning on COCO2017, every model was finetuned with the BCE with logits loss function

Model	Pretraining Dataset	Pretraining Setting	Epochs	Learning Rate	Optimizer	Micro F1
BiT-M-R50x1	ImageNet-21k (20M)	Supervised	15	0.1:reduce on plateau	SGD, m: 0.9, wd:1e-5	0.3237
ResNet50	ImageNet-1k (1M)	Supervised	15	0.1:reduce on plateau	SGD, m: 0.9, wd:1e-5	0.4039
ResNet18	ImageNet-1k (1M)	Supervised	15	0.1: reduce on plateau	SGD, m: 0.9, wd:1e-5	0.338
CLIP: ViT-B/32	400M images from web	Supervised	20	0.001	SGD, m: 0.9, wd:1e-5	0.3139
MoCo ResNet50	ImageNet-1k (1M)	Self Supervised	10	0.1: reduce on plateau	SGD, m: 0.9, wd:1e-5	0.3132

Table 4: Results from Finetuning on Open Images v4, every model was finetuned with the BCE with logits loss function

Using this methodology, each model is characterized with two sets of scores describing its biases in the feature representation space: intra-class variation in the pretrained and finetuned feature space for single classes, and embedding distance in the pretrained and finetuned space for pairs of classes. The former provides an understanding of the model's ability to represent different classes: i.e. the higher the cosine similarity for a given class of images, the lower the intra-class variation relative to other classes, and this shows us how well the model clusters a class relative to other classes. The embedding distance gives us a more direct representation of bias in examining the distance between two classes in the latent space. The higher the cosine similarity score, the closer the embedding of the two classes in the model's representation. For example, in the analysis sets, we can compare the distance between the classes car and car+man, and also car and car+woman. If the embedding distance between car and car+man is less than car and car+woman, it implies the model places car and car+man closer together in the embedding space, when car+man and car+woman should be equidistant from car. Furthermore, looking at the intra-class variation and embedding distance in both the pretrained and finetuned feature space, we test the impact of the following variables on how biases are represented: the dataset the model was pretrained on, the model architecture, the pretraining setting of the model, and the dataset the model was finetuned on. Systematically, we can discern which of these variables contributes to biases in the feature representation space by observing the change in the intra-cluster variation and the embedding distance in the pretrained and finetuned feature space. We test each of these variables by finetuning on COCO2017 and the Open Images Dataset.

We can analyze the results in two ways: by observing a model on its own, or by comparing across other models. By observing a model on its own, we can understand on well a model clusters a class of images in comparison to others by examining the intra-class variation metric and embedding distance metric scores. We can also observe how these scores change after finetuning the model on a dataset. If the scores for each class do not change significantly, we can attribute the biases from the model's architecture or the dataset it was pretrained on, overshadowing the biases coming from the dataset it was finetuned on. However, we cannot make any confident conclusions on whether the bias is originating from the dataset the model was finetuned on, the dataset the model was pretrained on, the network architecture or the training setting. In order to do this, we must systematically compare the biases across models by testing each of these variables one at a time. To test whether the bias originates from the dataset the model was finetuned, we can compare two models with the same architecture, pretrained on the same dataset with the same training setting and compare their pretrained intra-cluster variations and embedding distances, and their finetuned intra-cluster variations and embedding distances. In this way, we analyze each of the variables to get a better understanding of which factors contribute to biases in the feature representation space.

2.6 Results

To test the pretrained dataset, network architecture, and training setting, we will evaluate results from finetuning each of the models on the COCO2017 and Open Images Dataset. We evaluate the mean average precision score for finetuning on COCO2017 and the micro f1 score on Open Images for each of these models. The results and parameters are detailed in Tables 3 and 4. The features from the pretrained and finetuned models are evaluated using the intra-class variation metric and the inter-class distance metric defined in section Definition of Problem, using pairwise cosine similarity. We demonstrate our results using error bar plots where the middle of the error bar represents the average of the intra-class variation metric: $\{\mu_i\}_{i=0}^{n}$ for $i \in n$ or inter-class distance metric: $\{\mu_{ij}\}_{i=0}^{n}$ for $i \in n$ (where n is the number of classes in an analysis set) from T iterations, and the bars represent the min(T) and max(T): the range of values from repeatedly calculating the pairwise cosine similarity between folds of the features. The x-axis represents the classes in the analysis set, and the y axis represents the intra-class variation or inter-class distance scores. We also perform a one sample t test with the null hypothesis being that the mean is in fact the mean of the sampled data and found that all our results failed to reject this null hypothesis and as a result, the scores in the error bar plots are statistically significant. Note, the results for all analysis done with the Open Images Analysis set is in Appendix B.

2.7 Pretrained Dataset

As detailed in Table 3, ResNet50, Moco ResNet50, and ResNet18 have all been pretrained on ImageNet1K, Big Transfer ResNet50 has been pretrained on ImageNet21K, and the

Vision Transformer ViT/B-32 from the CLIP model has been pretrained on 400M images scraped from the Internet. We first examine the impact of these different pretrained datasets on the biases exhibited in the feature space before and after finetuning on COCO2017 and Open Images. To better compare the impact of the pretrained dataset on the biases, we only examine results from ResNet50, Big Transfer ResNet50, and ViT/32-B from CLIP. All these models have been trained in a supervised setting, and ResNet50 and Big Transfer ResNet50 have the same architecture so we can reasonably observe the impact of the pretrained dataset on the biases with respect to the other variables being constant. This is not the case for ViT/32-B which differs in its architecture however it is included here as it has been pretrained on 400M images from the Internet.

2.7.1 Pretrained Dataset: COCO 2017

We first examine results by finetuning ResNet50, Big Transfer ResNet50, and CLIP ViT/32-B on COCO2017 and analyzing the results on the COCO2017 analysis set. Each column in Figure 5 represents results of the intra-class variation metric, and inter-class distance metric in the Car, Refrigerator, Surfboard categories and each row represents results from the ResNet50, Big Transfer ResNet50, and CLIP ViT/32-B models in the pretrained feature space respectively. For categories like 'car+man', 'car', or 'woman', we use the intra-class variation metric to evaluate each error bar where the intra-class variation metric is calculated over a class of images for T iterations and the error bars represent the range of these values. For classes like 'car+woman vs car' where we are comparing two separate classes, we calculate the inter-class distance between 'car+woman' and 'car'. Intuitively, the 'random' category should have the lowest score since it has the greatest intra-class variation and thus the lowest cosine similarity score. The plots for ResNet50 and Big Transfer ResNet50 show us the impact of pretraining on ImageNet21K and ImageNet1K on the biases in the pretrained feature space. For example, we see that 'car+woman vs. car' has a lower cosine score than 'car+man vs. car' in Big Transfer ResNet50 and this is flipped in ResNet50. So this shows us that pretraining on ImageNet1K could result in ResNet50 exhibiting a stronger association between car+woman and car whereas ImageNet 21K may contain more of a bias towards car and man and thus these categories are represented closer together in the feature space. To generalize this analysis, we can compare the trends of these features across models. The results from Big Transfer ResNet50 and CLIP ViT/32-B show a very similar trend across features and thus we can conclude that ImageNet21K and the 400M images that ViT/32-B was trained on represent the categories for car, refrigerator and surfboard in a similar way. On the contrary, ImageNet1K exhibits different biases. For example, in the Surfboard category, ImageNet1K has a higher cosine score for the 'surfboard+man' class than the 'surfboard+woman' class, but this is flipped for ImageNet 21K and the 400M images from the web. These plots show us the impact of the pretraining dataset on the biases in the feature representation space where a similar trend *across* models implies the datasets the model was pretrained on contain similar biases.

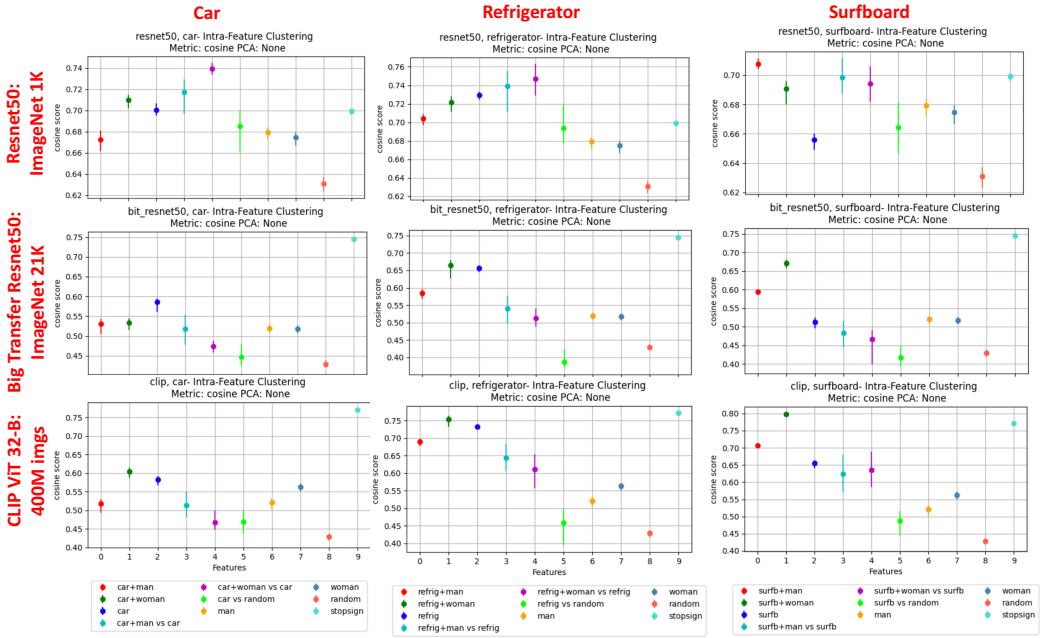

Figure 5: ResNet50, Big Transfer ResNet50, and CLIP ViT/32-B: intra-class variation and inter-class distance for classes of interest in the COCO2017 analysis set: Car, Refrigerator, Surfboard. A similar trend across CLIP ViT/32-B and Big Transfer 21K implies that both of these datasets encode and reflect similar biases in the pretrained space. However, the model architecture could potentially be impacting the way these biases are represented. Instead, we can directly compare Big Transfer ResNet50 and ResNet50 since they use the same backbone architecture and we observe differences in biases across these two different pretraining datasets on the same model architecture.

We can look at the same three models in Figure 6, but instead, the first two columns represent the scores from the intra-class variation metric and the third column represents scores from the inter-class distance function. These plots are representing the same results, but grouped with different classes in each column. We examine the feature representations after finetuning on COCO2017 and calculate the spearman coefficient between the pretraining and finetuning scores. A similar trend between pretraining and finetuning scores results in a higher Spearman's coefficient as detailed in Big Transfer ResNet50 for the individual category plot. A similar trend between the pretraining and finetuning scores shows that the model mostly preserved it's biases from pretraining and was not greatly impacted by the biases after finetuning. This can be observed in the individual category plots and the comparison plots where we calculate the inter-class distance between two classes. However, this does not hold true for the paired plot for ResNet50. These plots show us that ImageNet21K and the 400M images from the web still influence the biases even after the model has been finetuned on COCO2017, but this is less true for models pretrained on ImageNet1K since the biases seem

to shift more after finetuning on COCO2017 which implies that the biases in the finetuned feature space were potentially impacted by the biases in the COCO2017 dataset.

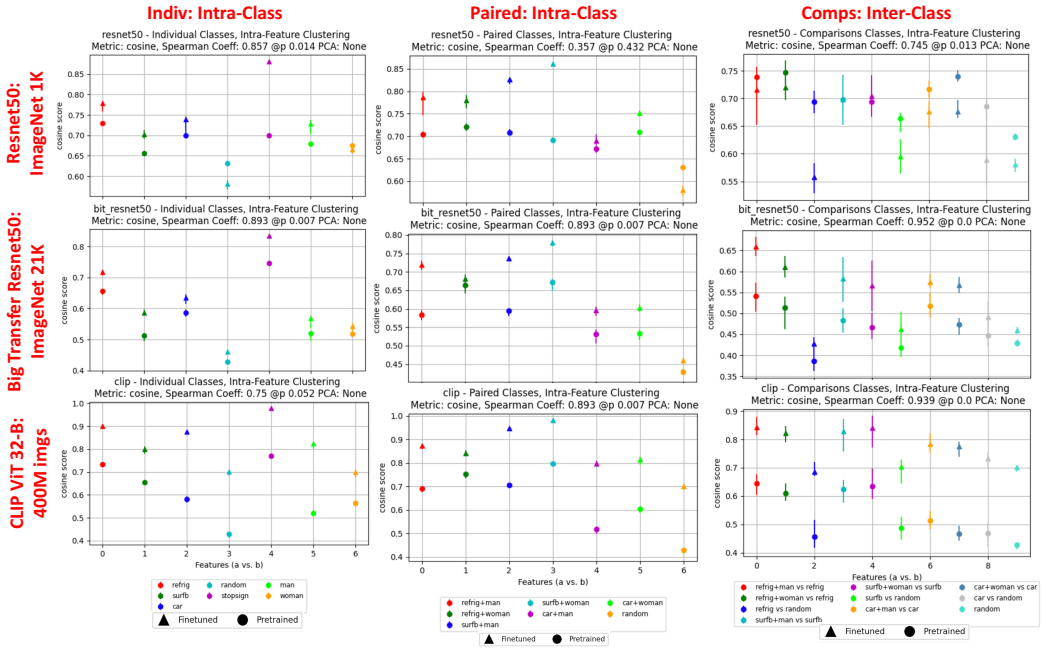

Figure 6: ResNet50, Big Transfer ResNet50, and CLIP ViT/32-B: these plots segment the categories in the COCO2017 analysis set by individual classes (car, refrigerator, surfboard, etc.), paired classes (refrig+man, surfb+woman etc.), and comparisons (refrig vs. random). These plots show the results of the intra-class variation metric and the inter-class distance metric of each class before and after finetuning. A similar trend before and after finetuning implies that the model preserved its biases from finetuning whereas a different trend implies that finetuning impacted the biases.

More concretely, we can plot the pretraining and finetuning scores on a single plot and directly examine their correlation as shown in Figure 7. A linear correlation implies that the trends were preserved from pretraining to finetuning whereas a non-linear curve implies that finetuning impacted the model's biases. Similar to the conclusions made in Figure 6, we see that ResNet50's paired plot shows the most non-linearity between the pretrained and finetuned features. Thus, we can conclude that COCO2017 impacted the biases of the paired categories in the analysis set for models trained on ImageNet1k. To examine how these biases were impacted, we can observe the plots in Figure 6.

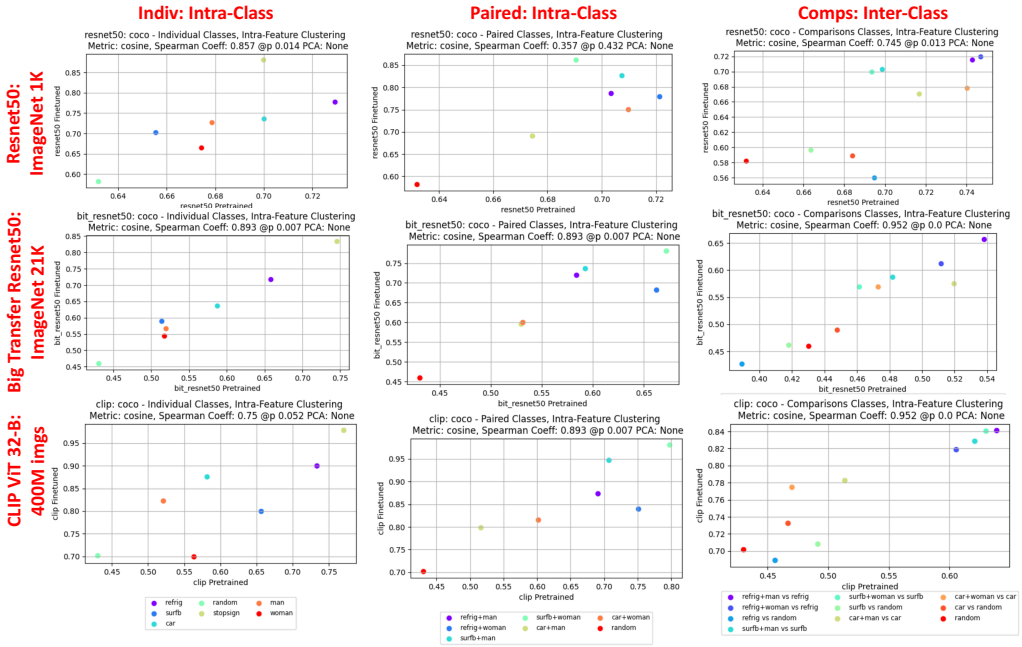

Figure 7: ResNet50, Big Transfer ResNet50, and CLIP ViT/32-B: Plots the trend between the pretrarining and finetuning scores.

2.8 Network Architecture

To examine the impact of the network architecture on the biases. We choose to look at ResNet50, ResNet18, and ViT/B-32. Both ResNet50 and ResNet18 were pretrained on the same dataset and all three models were trained in a supervised setting.

2.8.1 Network Architecture: COCO 2017

We observe the same format of the plots Section 2.7. We can compare the trends of the pretrained scores in Figure 8 across models and observe that ResNet18 and ResNet50 exhibit similar biases in the pretrained feature space. This is not the case for for Clip ViT/B-32. From this, we can conclude that despite being a different network architecture, ResNet18 and ResNet50 encode biases in a similar way. However, this is not the case for CLIP ViT/B-32. We cannot confidently conclude that the difference in these biases originates from the network architecture because CLIP ViT/B-32 was pretrained on a different dataset than ResNet50 and ResNet18. However, we do not have access to CLIP ViT/B-32 pretrained on ImageNet and thus we can only conclude that either the pretrained dataset and/or the model architecture of CLIP ViT/B-32 results in different biases in the pretrained feataure representation space.

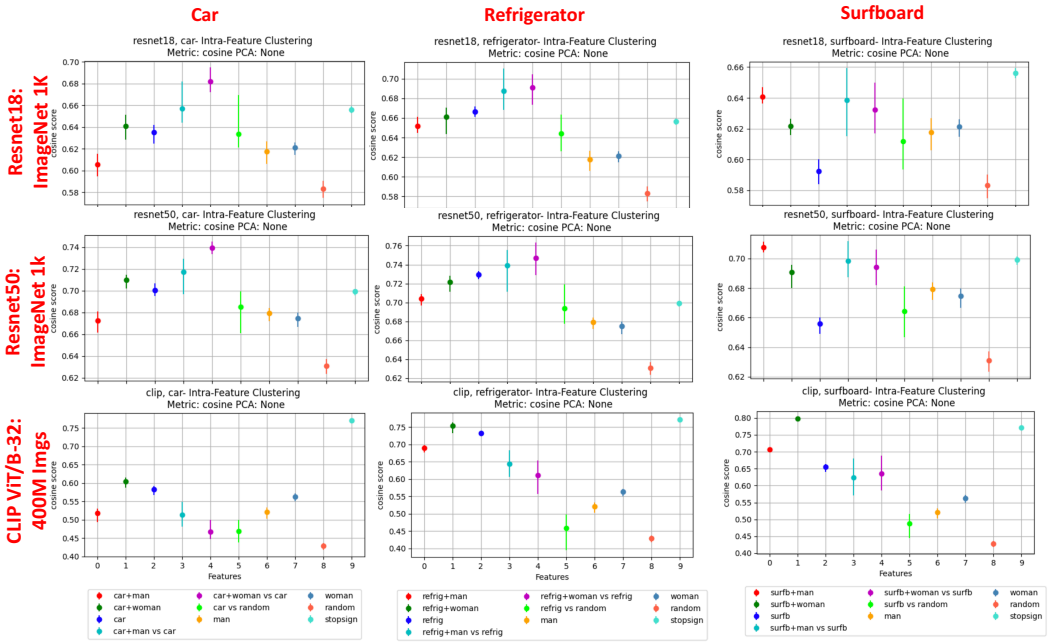

Figure 8: ResNet18, ResNet50, CLIP ViT/B-32: Analysis of biases in the pretrained feature space of categories in the COCO2017 analysis set with respect to network architecture. Similarities in the trend across ResNet18 and ResNet50 imply that the network architecture did not have an impact on how these biases were encoded in the pretrained feature space. However, this is not true for CLIP ViT 32-B. We cannot conclude whether the difference in biases for CLIP ViT 32-B is due to the architecture or the pretrained dataset, we can just conclude that the biases are different than ResNet50 and ResNet18.

We can observe these biases after finetuning as well in Figure 9. For both ResNet18 and ResNet50, the trends for the individual class plots and the comparison classes are similar and the trends for the paired classes plot after finetuning diverges in the same way for both models. This implies that ResNet18 and ResNet50's biases were impacted after finetuning on COCO2017 and their trends after finetuning are similar implying that their biases were impacted in a similar way after finetuning. However, for CLIP ViT/B-32, the trends before and after finetuning are mostly preserved implying that it was not impacted by finetuning on COCO2017. From this analysis, we can conclude that despite different network architectures, the biases were similar for ResNet18 and ResNet50 and so the dataset that the model was pretrained on has more of an impact on the biases in the pretrained and finetuned space than the network architecture.

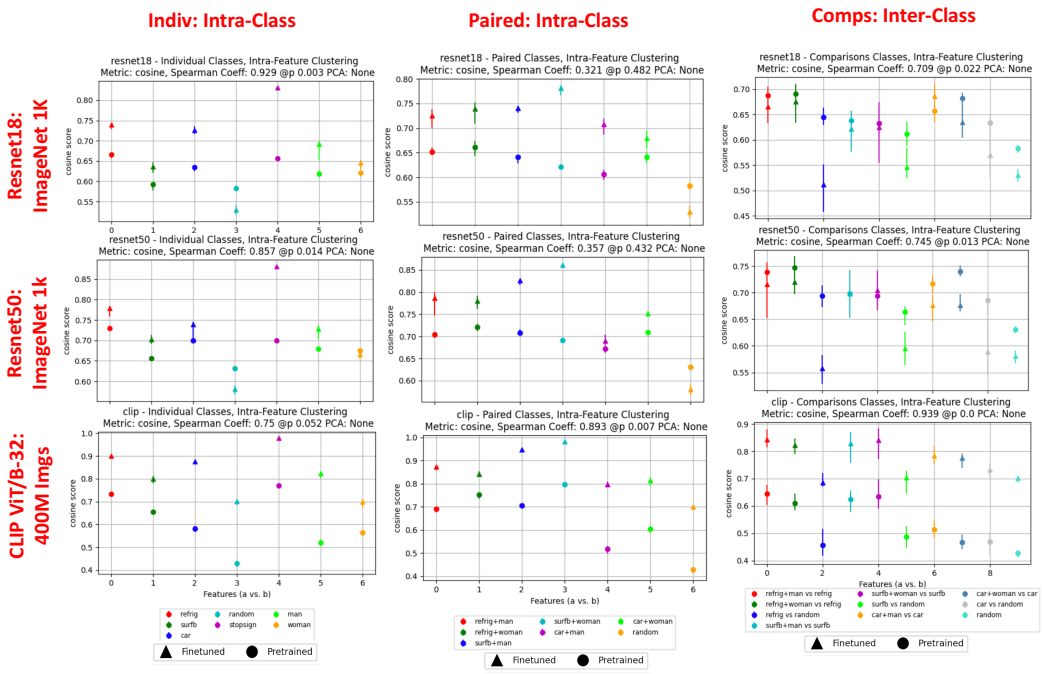

Figure 9: ResNet18, ResNet50, CLIP ViT/B-32: Analysis of biases before and after finetuning on the COCO2017 analysis set. Low spearman coefficients for the paired ccategoreis in ResNet18 and ResNet50 imply that both of these models absorbed biases from the finetuning dataset since the trend diverges from the pretrained feature scores. A similar trend for the rest of the models shows that the biases for those classes was mostly preserved.

Observing the same phenomenon in Figure 10, we can compare the trends between the pretraining and finetuning scores for each model. The linear trends in all three of the models implies that finetuning did not impact the biases of the pretrained model for the individual classes and the comparison classes. However, this is not the case for the paired classes for Resnet18 and Resnet50. This is also what we observed in Figure 9

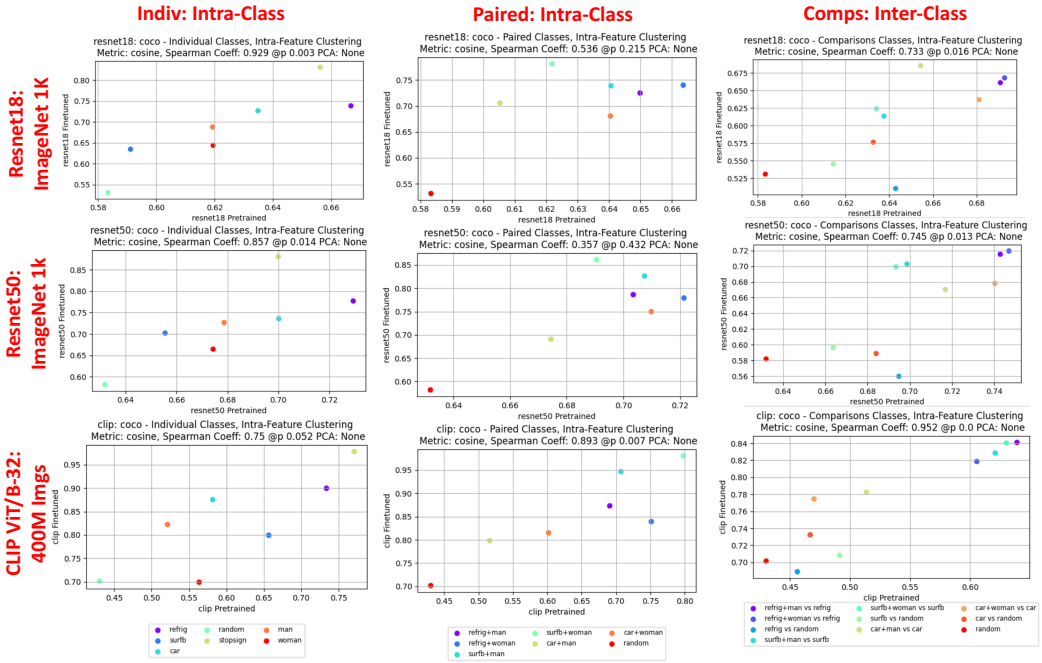

Figure 10: ResNet18, ResNet50, CLIP ViT/B-32: Trends between pretraining and finetuning scores. We observe a less than linear trend for the paired classes for ResNet18 and ResNet50 implying that these paired classes were impacted by biases in the finetuning dataset.

2.9 Training Setting

We compare MoCo ResNet50 and ResNet50 to examine how the training setting impacts biases. MoCo ResNet50 was pretrained in a self supervised setting whereas ResNet50 was pretrained in a supervised setting. Both of these models have the same backbone architecture and have been pretrained on the same dataset.

2.9.1 Training Setting: COCO 2017

Figure 11 compares MoCo ResNet50 and ResNet50. There is a clear difference in trends between the biases in the pretrained feature representation space. With all other variables constant (network architecture, and pretraining dataset), we can reasonably assume that these differences in biases are due to the training setting. For example, ResNet50 has a higher score for 'car+woman vs car' than 'car+man vs car' and 'car' whereas this is flipped for MoCo ResNet50 implying that MoCo ResNet50 placed 'car' and 'car+man' closer in the embedding space than 'car' and 'car+woman'. This difference can be attributed to the pretraining setting.

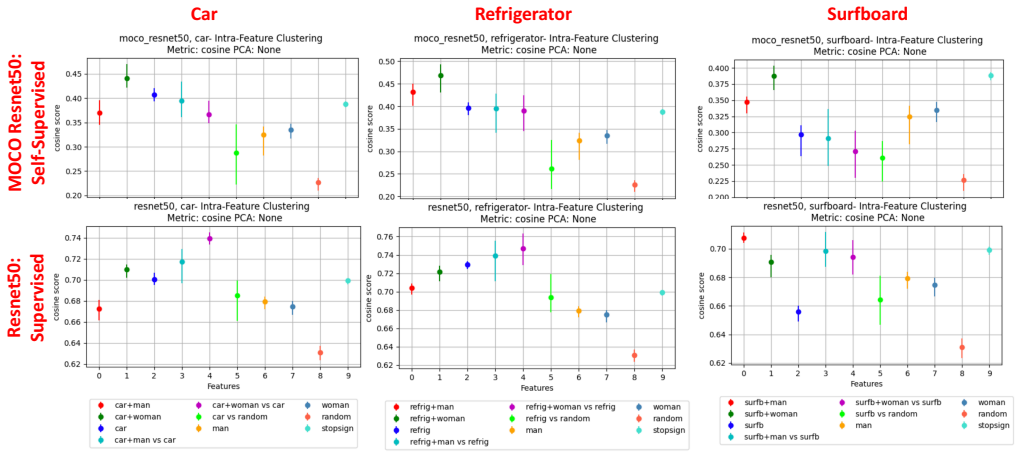

Figure 11: ResNet50, MoCo ResNet50: Analysis of biases in the pretrained feature space of categories in the COCO2017 analysis set. The different trends between these two models imply that the training setting impacted the way these classes are represented in the feature space.

Figures 12 and 13 show the impact of finetuning on COCO2017. For the paired classes, ResNet50 does not have a linear association between the pretrained and finetuned scores. This implies that it was impacted by the biases in the COCO2017 dataset. For the rest of the results, we observe that MoCo ResNet50 and ResNet50 mostly adhere to the biases in their pretrained feature space implying that different supervised settings do not impact biases of a model after finetuning on COCO2017.

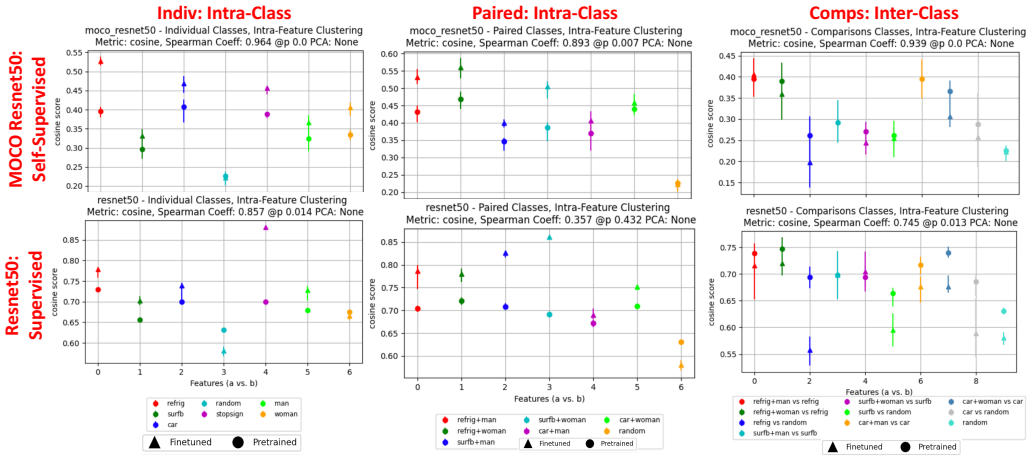

Figure 12: ResNet50, MoCo ResNet50: Analysis of biases before and after finetuning on the COCO 2017. A less linear trend for the paired classes for ResNet50 implies that a supervised setting may impact the biases in the finetuned feature space of the paired classes more than an unsupervised setting.

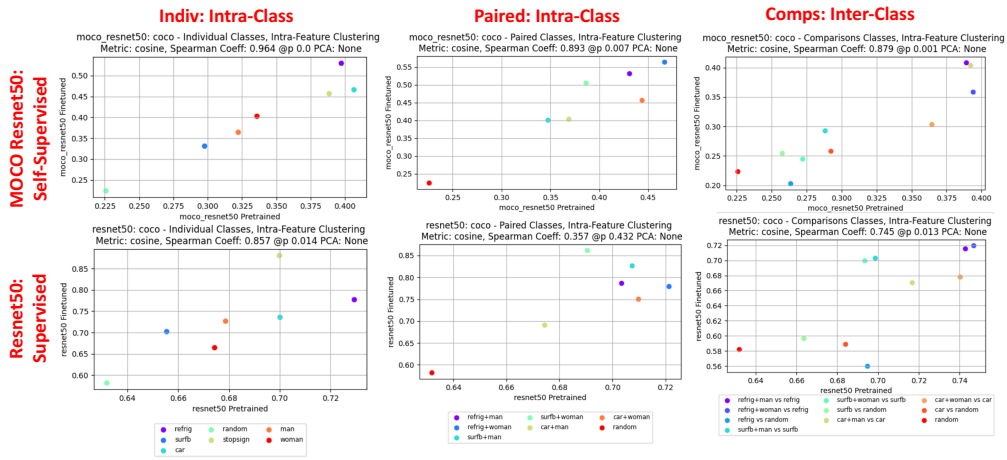

Figure 13: ResNet50, MoCo ResNet50: Trends between pretraining and finetuning scores.

2.10 Comparison Across Models

We can also examine pairs of models by comparing their pretrained and finetuned trends. The pretrained trends show us a comparison of the model's biases with respect to its network architecture, its pretraining setting and its pretraining dataset. The comparison of the finetuned trends show us the impact of finetuning the models on COCO2017 or Open Images. If the pretrained trends for two models are similar, it results in a linear trend and shows that the biases of the pretrained models were similar. However, we cannot conclude which factors contributed most to those similar biases just by looking at these trend plots since the variables are not necessarily constant and there could be a range of factors affecting these differences of biases. If the finetuned trends are the same and result in a linear trend line, we can reasonbly assume the biases originated from the dataset that the model was finetuned on given all other variables of the models are the same. However, if this trend is less than linear, it implies that at least one of the two models' biases in the finetuned space was impacted by it's biases in the pretrained feature space. This would require examining the models individually in the sections above to compare how a model's biases change from pretraining to finetuning and which factors contributed most to the changes in a model's biases.

2.10.1 Model Comparison: COCO2017

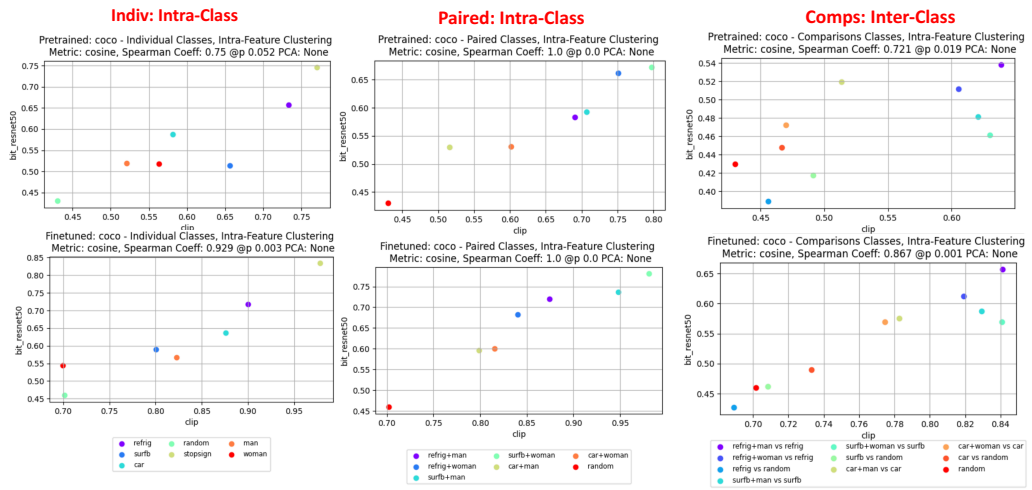

Figure 14: Comparison of CLIP ViT/B-32 and Big Transfer ResNet50. The top row compares the pretrained trends *across* these two models whereas the bottom row compares the finetuned trends. Since the trends after finetuning for both of these models are more linear (higher spearman coefficient) than their pretrained trends, we can conclude that the biases in the finetuned feature space came from the dataset that the model was finetuned on.

Comparing CLIP: ViT/B-32 and Big Transfer ResNet50, it's clear that the trend for the paired classes is linear. This indicates that for paired classes, both of these models encoded their biases similarly before and after finetuning. However, we cannot make any conclusions regarding which variables contributed most to the bias because there are no controlled variables across these two models.

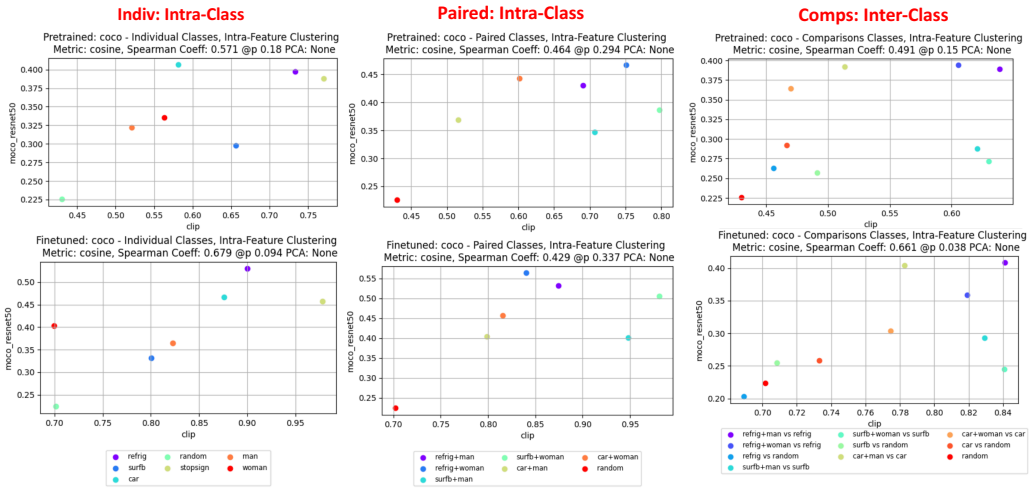

Figure 15: Comparison of CLIP ViT/B-32 and MoCo ResNet50. The pretrained trends are not linear implying that these two models encode their biases very differently in the pretrained feature space. However, these trends become more linear after finetuning implying that they both absorbed and reflect some biases from finetuning on COCO 2017. To better examine where these biases came from and how they are represented for each of the models, we would have to examine each model's biases independently as done in the sections above.

Comparing CLIP ViT/B-32 and MoCo ResNet50, the trends across all plots are not linear. This implies that these two models encode their biases very differently before and after finetuning. Furthermore, they differ in their network architecture, training setting and pretraining dataset. As a result, we cannot conclude which factors contributed most to these differences in biases.

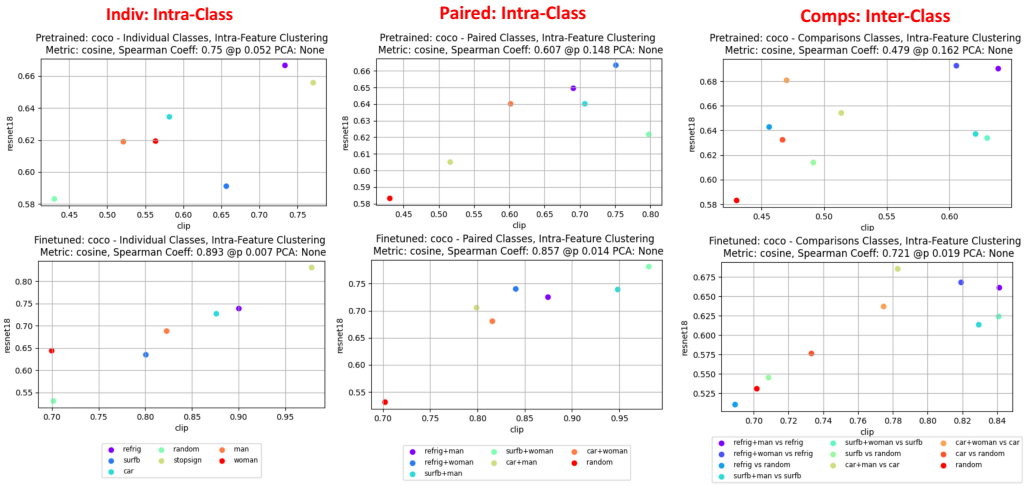

Figure 16: Comparing CLIP ViT/B-32 and ResNet18. The trends after finetuning are more linear than the pretraining trends implying that the models absorbed biases from the dataset after finetuning.

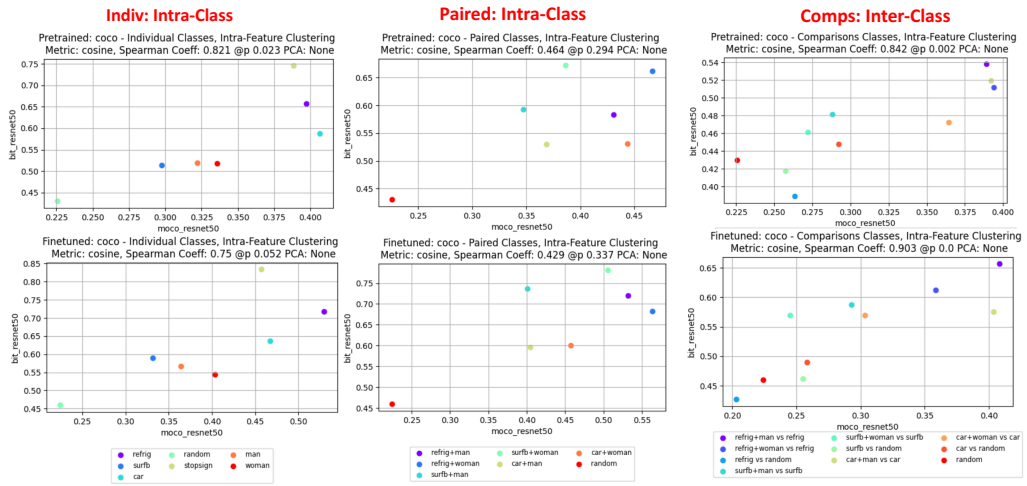

Figure 17: Comparing MoCo ResNet50 and Big Transfer ResNet50, the biases in the pretrained space were similar for the individual and comparisons categories but they became less similiar after finetuning for the individual classes. This could imply that finetuning impacted these models differently. For example, one of the models might not have been impacted by finetuning resulting in a less than linear trend.

Comparing MoCo ResNet50 and Big Transfer ResNet50, the trends for the individual classes and the comparison classes are close to linear implying that these models encode biases similarly despite being pretrained with a different pretraining setting. This implies

that the pretraining setting may not have a big impact on the representation of biases for the ResNet50 architecture.

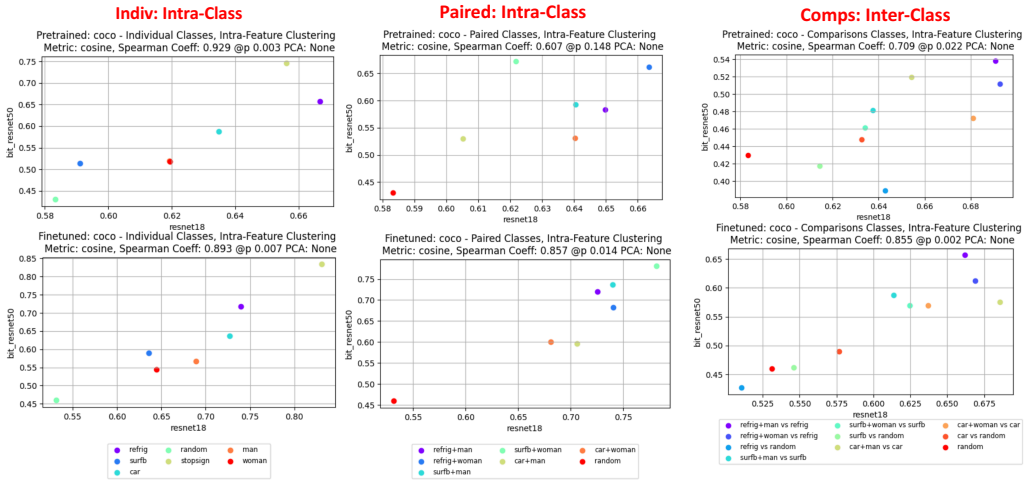

Figure 18: Comparing ResNet18 and Big Transfer ResNet50. The trends for the paired plots and the comparison plotos become more linear after finetuning implying that both of these models were impacted by the biases of the finetuning dataset. However, this is not true for the individual categories. Although the trend is still close to linear, it could imply that one of the models was impacted less by the biases in the finetuning dataset.

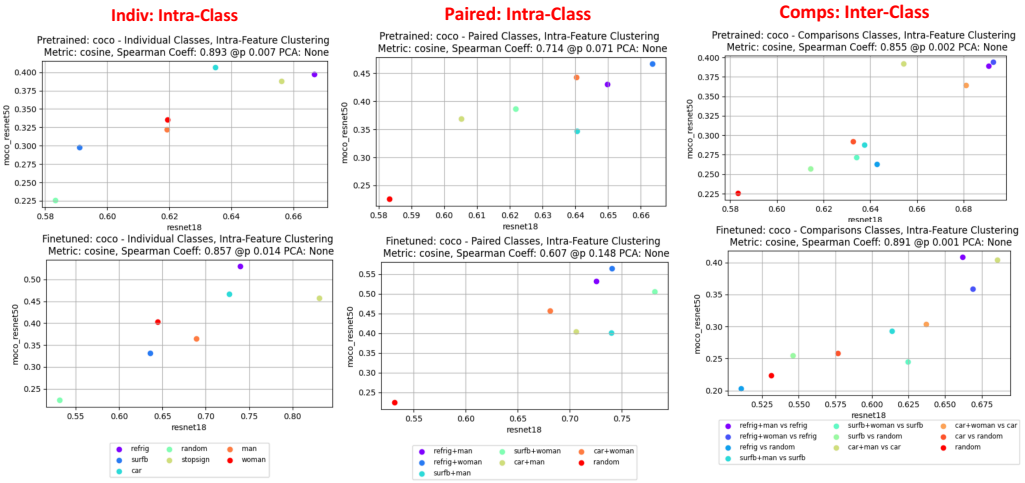

Figure 19: Comparing ResNet18 and MoCo ResNet50. The linearity of the trends is similar before and after finetuning. For the paired categories, the correlation decreased implying that one of the models was impacted differently by finetuning on COCO2017.

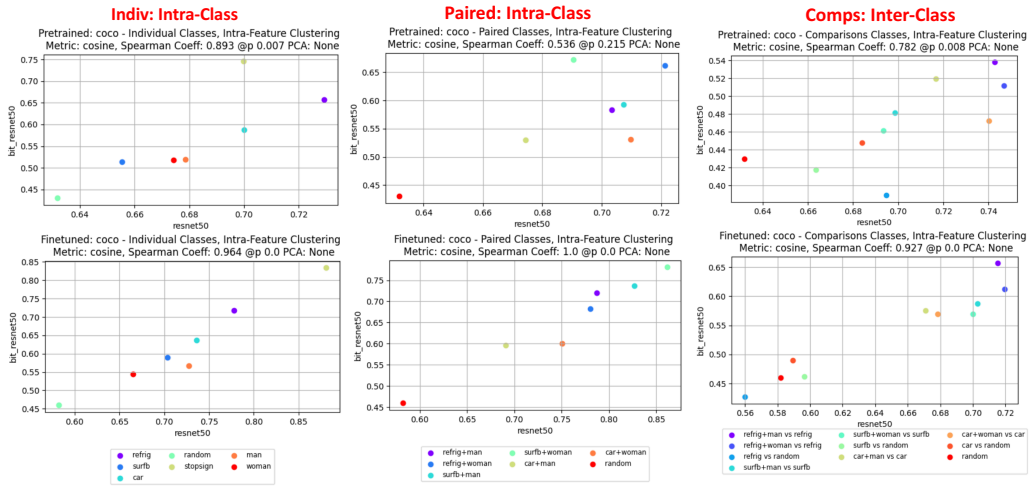

Figure 20: Comparing ResNet50 and Big Transfer ResNet50. The trend become very close to linear after finetuning implying that both of these models were impated by finetuning in a very similar way even though their biases in the pretrained space were different.

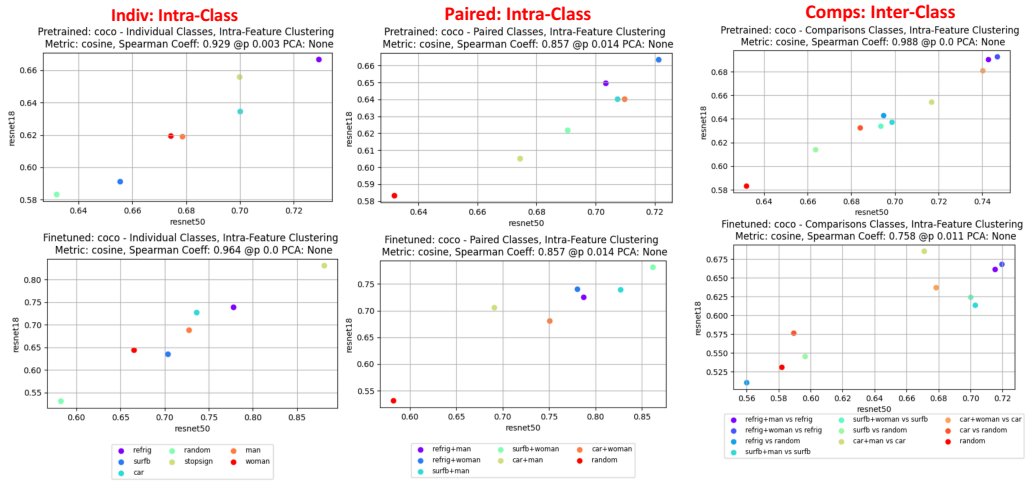

Figure 21: Comparing ResNet50 and ResNet18. These trends before and after finetuning are also close to linear implying that their biases in the pretrained and finetuned space are encoded in a similar way. It would be necessary to examine the biases of each model independently to examine how the biases are represented.

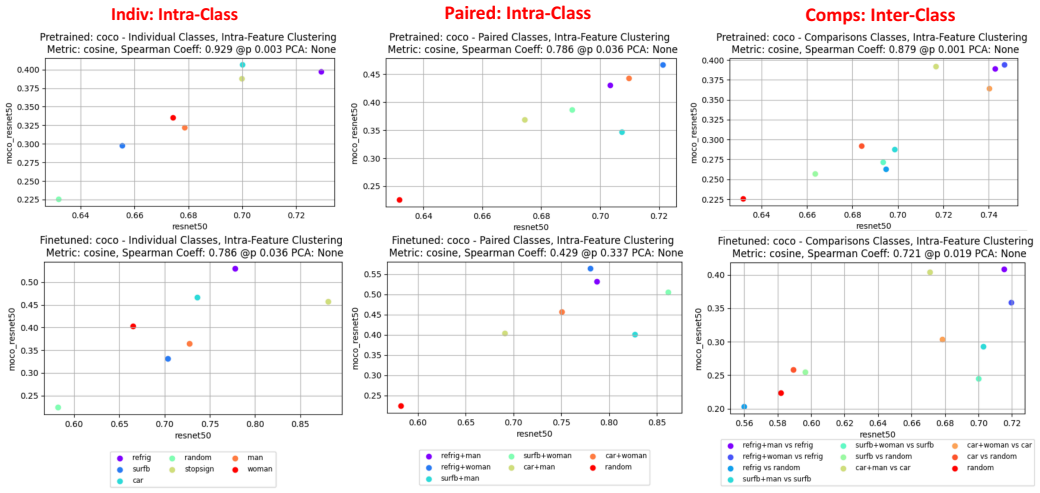

Figure 22: Comparing ResNet50 and MoCo ResNet50. The trends become less linear after finetuning implying that they may encode their biases in a similar way but they diverge after finetuning.

2.11 Discussion

In sections Pretrained Dataset Network Architecture, and Training Setting, we test three variables: pretrained dataset, network architecture, and training setting for their effects on biases before and after finetuning on the COCO2017 and Open Images Dataset. By holding other variables constant, we can discern which of these variables contributed most to changes in the model's biases after finetuning. We further observed that by comparing the trends between the pretraining feature scores and finetuning feature scores for a single model, we can observe if the model preserved biases from its pretrained weights. Similarly, in section Comparison Across Models, we can observe these same trends *across* models to compare how different models encode biases in their pretrained and finetuned feature space. This analysis provided a qualitative methodology to compare the biases across two models with respect to their pretrained weights, and compare how these biases shifted after finetuning on a dataset. However, we cannot discern where these biases originated from without examining the models individually as was done in sections Pretrained Dataset Network Architecture, and Training Setting. We use our definition of bias to qualitatively compare the relative trends of the intra-class variation and embedding distance for a class of images in an analysis set. This provides us with a high level, interpretable methodology for comparing biases across models without the need for a metric that generalizes across different models' latent representations, This study could benefit from additional models for comparison. However, one of the biggest limitations is training these models to a reasonble, comparable performance on datasets such as Open Images. This is because Open Images has 601 classes and is a very large dataset requiring a lot of hyperparameter tuning in order to get reasonable performance. As a result, a natural future direction for this work would be to incorporate more models for comparison

so we can perform more controlled experiments with respect to the variables we explored in this study.

3 Summary

In this work, we explore gender biases in the feature representation space of visual recognition models. We aggregate subsets of the COCO2017 and Open Images v4 datasets containing images with objects that co-occur with gender so we can study how models represent gender in the feature space. We identify metrics that capture gender biases in the feature space and propose a methodology that identifies factors which contribute to a model's biases. More specifically, we test how the pretraining dataset, finetuning dataset, training setting and network architecture impact the model's biases in the pretrained and finetuned feature space. Using this methodology, we are able to qualitatively compare biases across models and examine which models are more susceptible to biases with respect to the variables we test. This work serves as an extension to modeling biases in natural language processing settings. Past studies [13] have examined bias by measuring the distance between word embeddings in the latent space. We extend this idea to the image domain and examine the distance between class representations, and we analyze the representation of the class itself. Unlike previous studies, we analyze biases at the class level instead of looking at individual images or instances of bias. This generalization serves as a vital first step in characterizing and comparing models for biases.

4 Future Directions

One of the challenges of bias studies is the availability of a labeled dataset that reflects the biased relationship of interest. One of the future directions of this work is to aggregate analysis sets with less noise. As detailed in Table 1, the COCO2017 analysis set has a small number of examples per class. This dataset was carefully curated by human inspection and thus is not reasonably scalable. Furthermore, COCO2017 does not have gender labels making it harder to easily collect large amounts of data for the study proposed in this paper. This can be solved by using datasets such as Open Images. However, as shown in the Appendix, the Open Images analysis set has a lot more noise since this set was not manually collected and instead, we relied on the labels in the dataset. Particularly for categories that contain a gender and an object (i.e. car+woman), we want to ensure there are no other objects or genders present in the image. However, this is very difficult to enforce in noisy datasets like Open Images. To reduce noise, we would need to ensure only the gender and the object are present in the classes with careful filtering and human inspection. The study in this paper used cosine similarity as the primary metric to calculate the similarity between two sets of features. However, cosine similarity does not capture the complex non-linear relationships that are often reflected in the feature space. With the subtle gender bias associations we are

trying to capture, we want to explore metrics that are better able to capture this non-linearity such as distance correlation and Mahalanobis distance. Furthermore, we want to test with other, standard interpretable metrics such as Euclidean distance. Due to the non-linearity of the latent space, we also want to experiment with principal component analysis. As detailed in the Appendix, there is a lot of variation in the analysis set. Furthermore, we are currently using every single feature dimension to represent a class of images. Principal component analysis could be beneficial in reducing the number of features and could better capture the geometry of the subspace. In terms of the experiments, we want to test other self-supervised models such as SimCLR [26] as another model for comparison with MoCo ResNet50. In terms of training settings, it would also be beneficial to explore how training a model from scratch impacts the biases in comparison to models that are pretrained and then finetuned.

In this work, we limited our gender biases to two genders and have yet to perform a more extensive qualitative analysis on the images themselves to understand the kinds of images that contribute most to bias in the feature representation space. In the future, we hope to extend this study to other genders beyond men and women with the increasing availability of labeled datasets for bias. Furthermore, it is necessary not only to observe who is represented in the bias relationships, but also how these genders are represented. For example, if we find a model to place sports and sports+woman closer together than sports and sports+man but all the images of sports+woman have sexualized representations of women, it would be necessary to perform a more thorough qualitative study of our analysis sets and examine how the genders are being represented more carefully. Moving forward, we want to further explore how we can develop improved metrics for representing biases in visual recognition models, and explore these feature representations at different points in the network's architecture to understand how biases propagate through the network. This work serves as a necessary exploratory step in characterizing biases at the class level in the feature representation of a model.

A Analysis Sets

A.0.1 COCO2017 Analysis Set

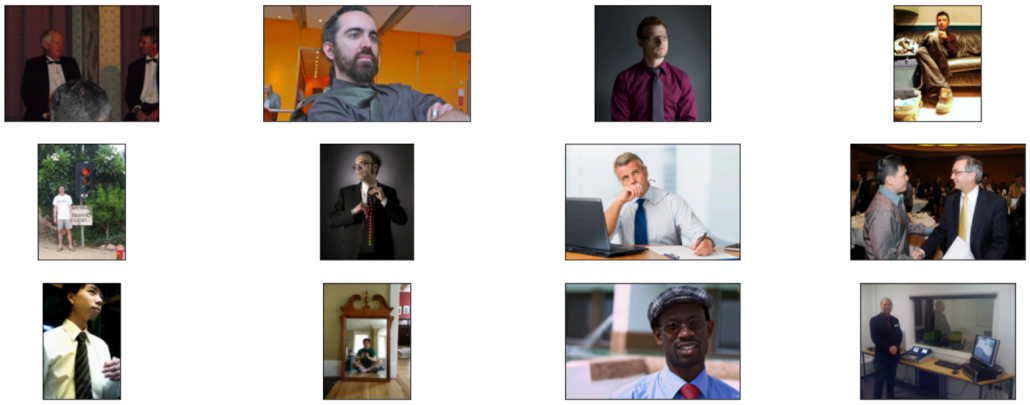

Figure 23: Man

Figure 24: Woman

Figure 25: Random

Figure 26: Stop Sign

Figure 27: Car

Figure 28: Car+man

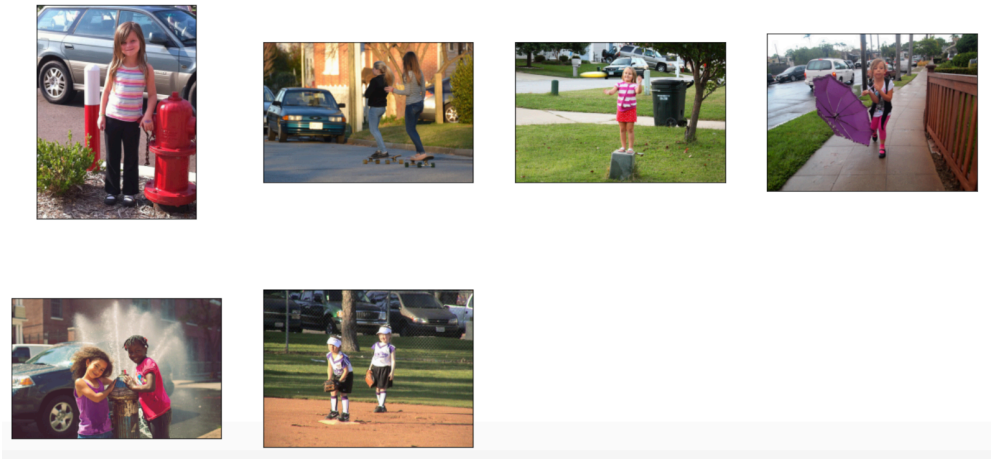

Figure 29: Car+woman

Figure 30: Refrigerator

Figure 31: Refrigerator+man

Figure 32: Refrigerator+woman

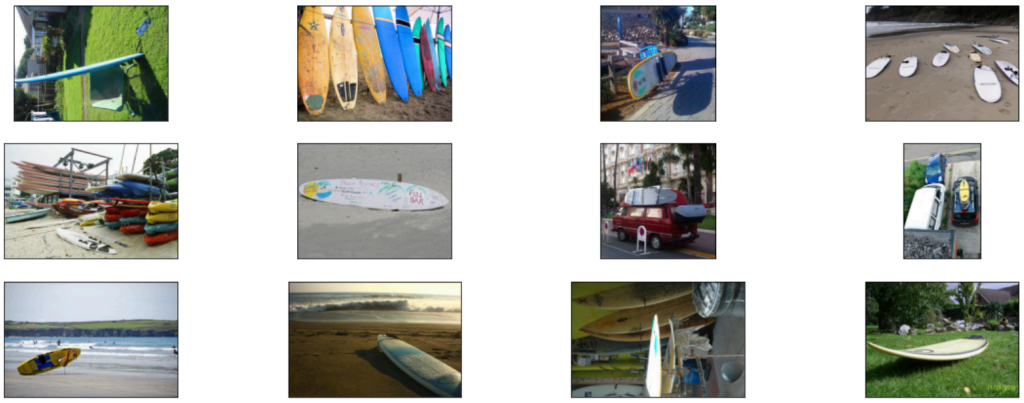

Figure 33: Surfboard

Figure 34: Surfboard+man

Figure 35: Surfboard+woman

A.0.2 Open Images v4 Analysis Set

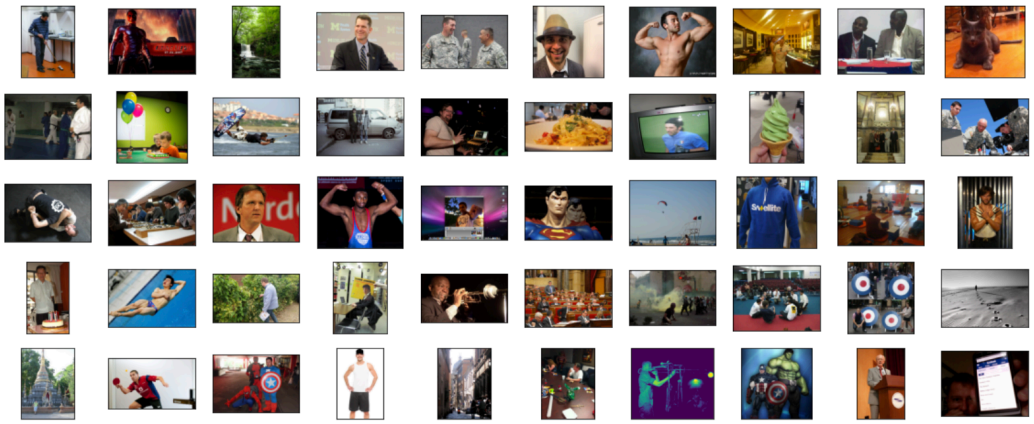

Figure 36: Man

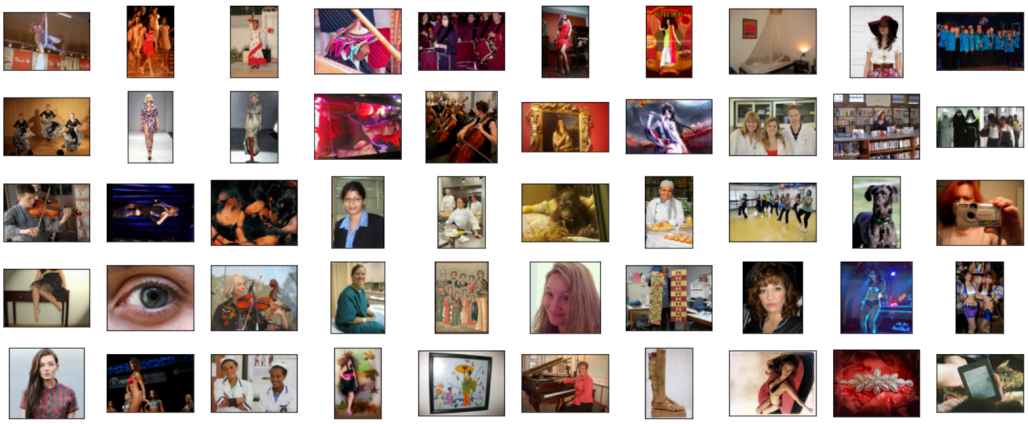

Figure 37: Woman

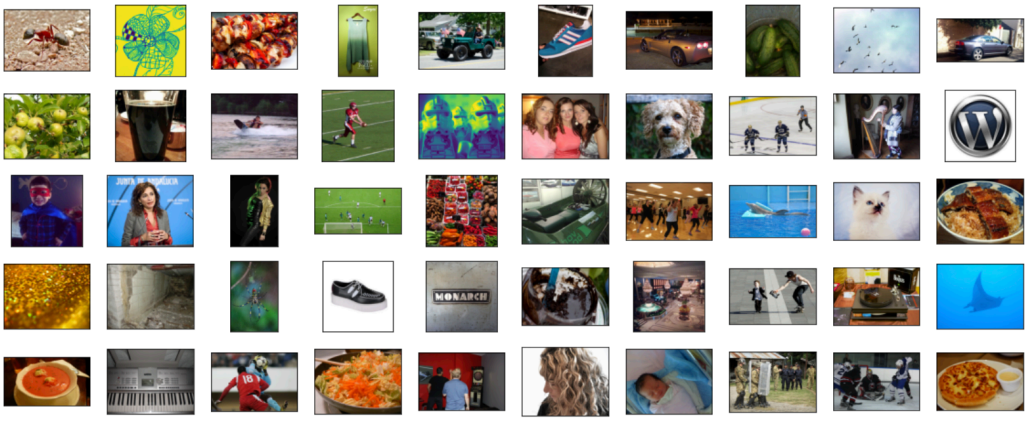

Figure 38: Random

Figure 39: Stop Sign

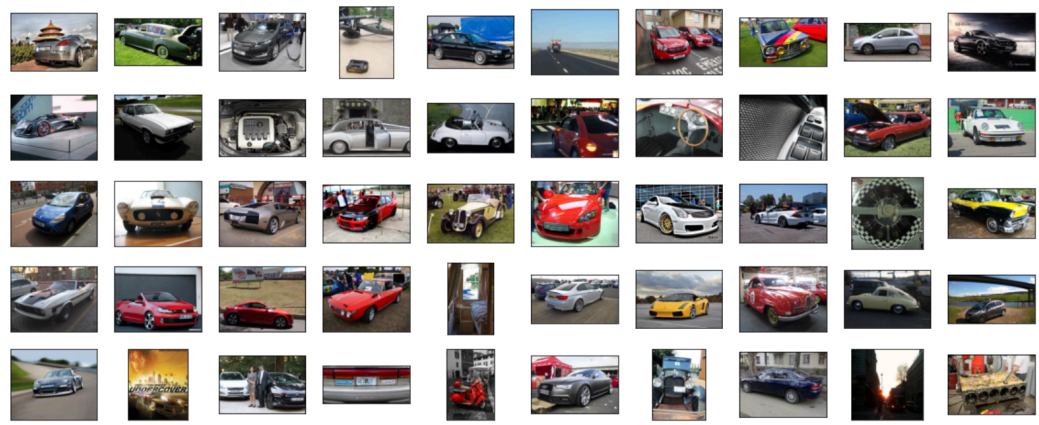

Figure 40: Car

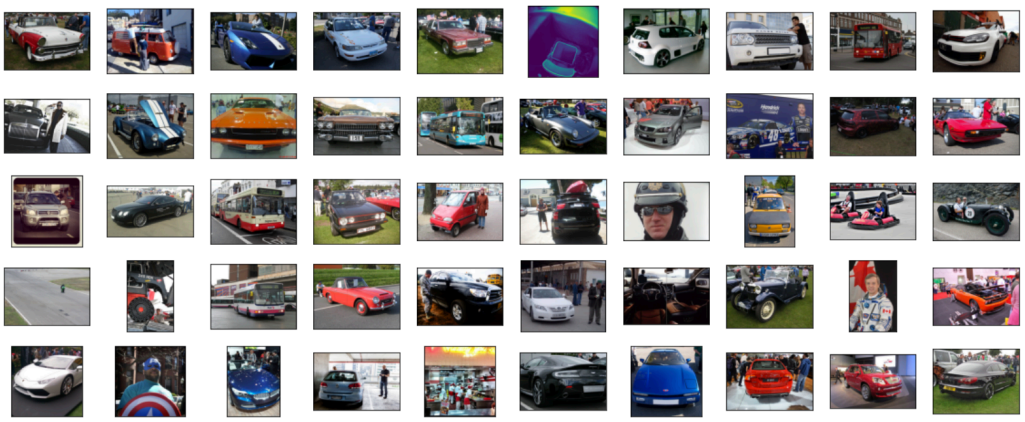

Figure 41: Car+man

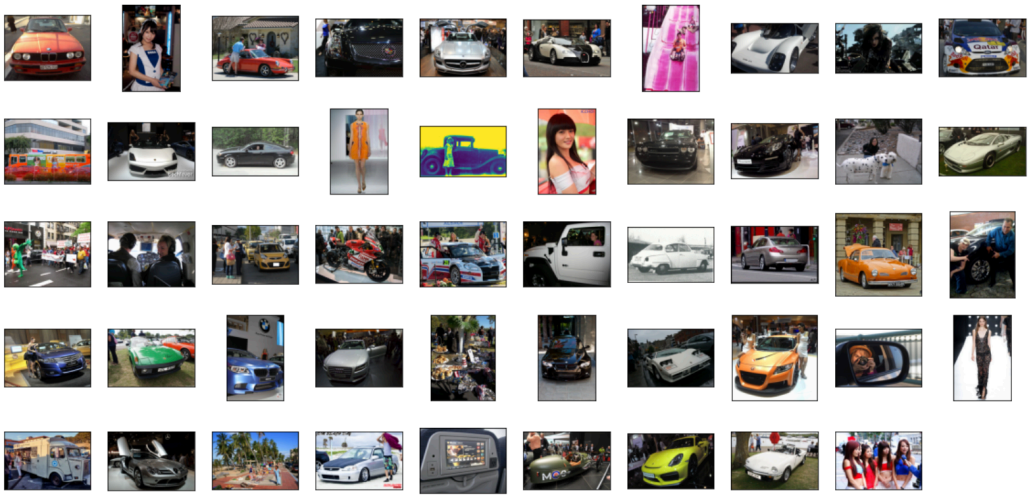

Figure 42: Car+woman

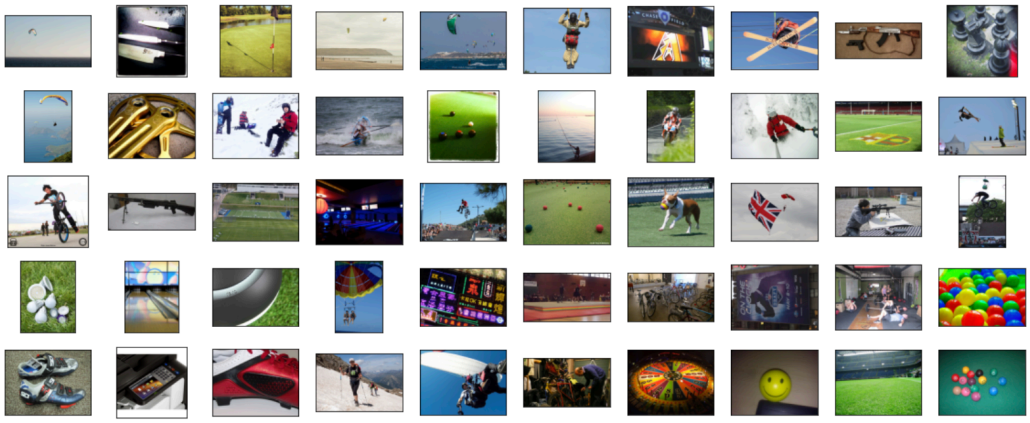

Figure 43: Sports Equipment

Figure 44: Sports+man

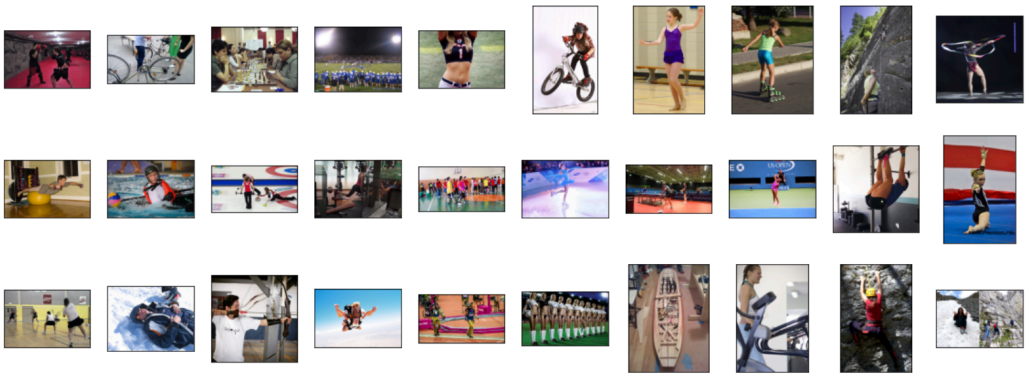

Figure 45: Sports+woman

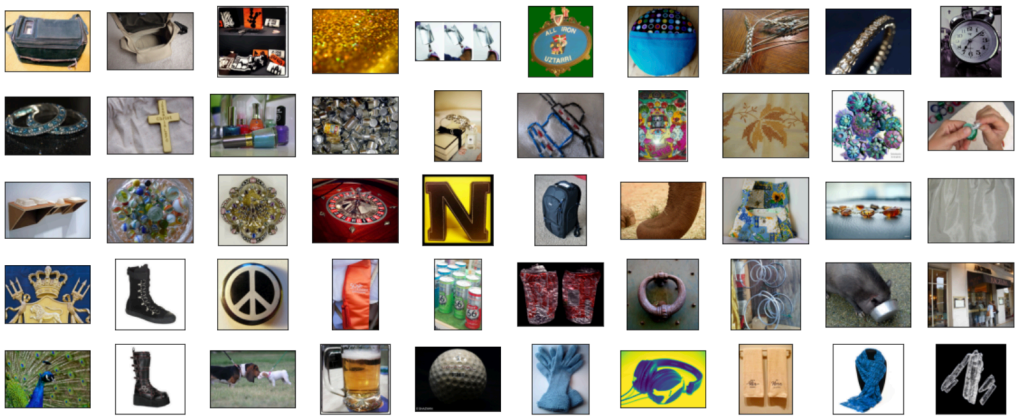

Figure 46: Fashion

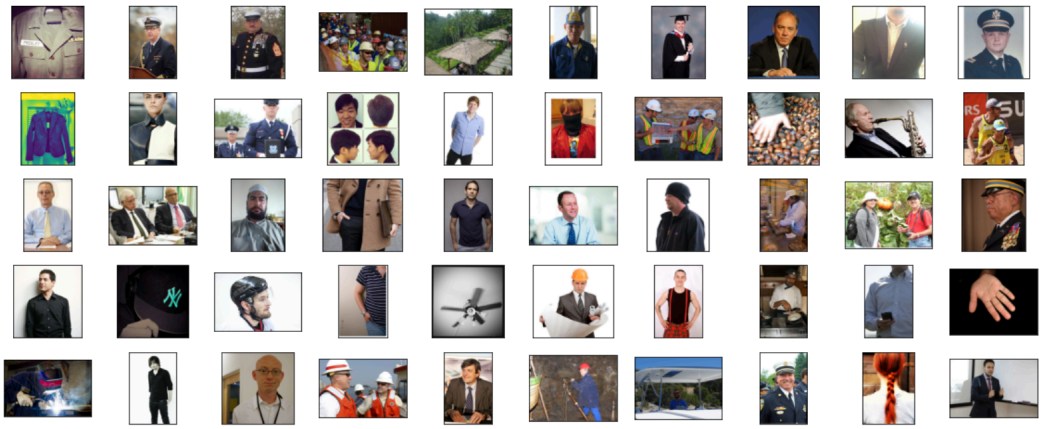

Figure 47: Fashion+man

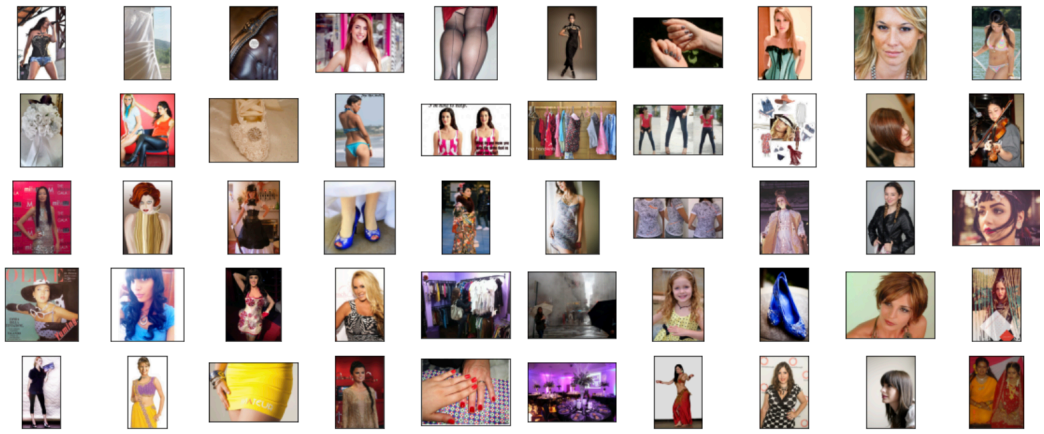

Figure 48: Fashion+woman

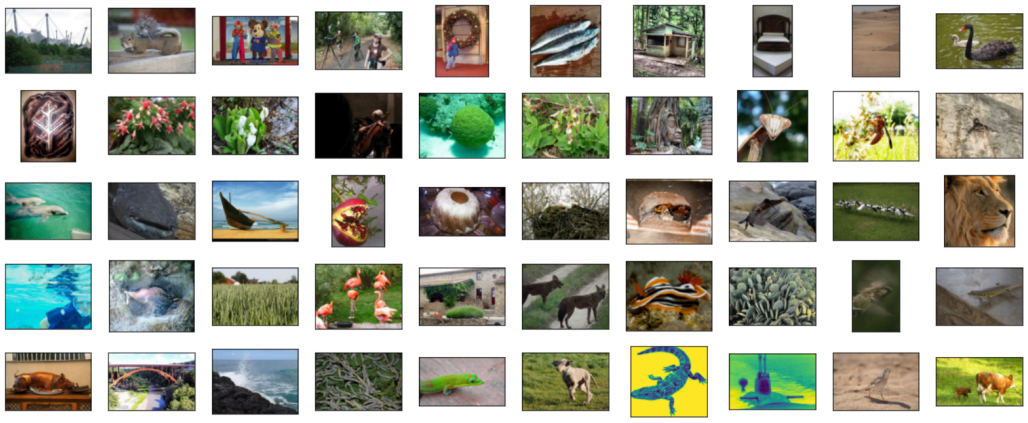

Figure 49: Mammal

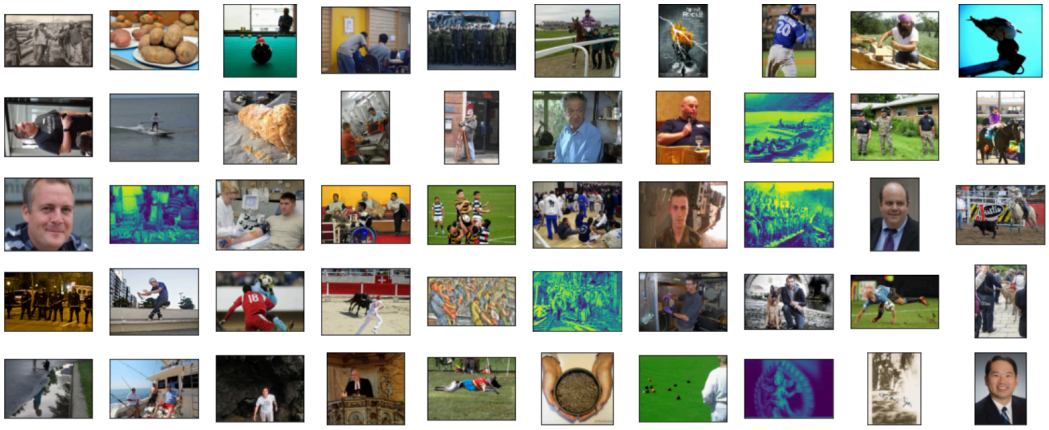

Figure 50: Mammal+man

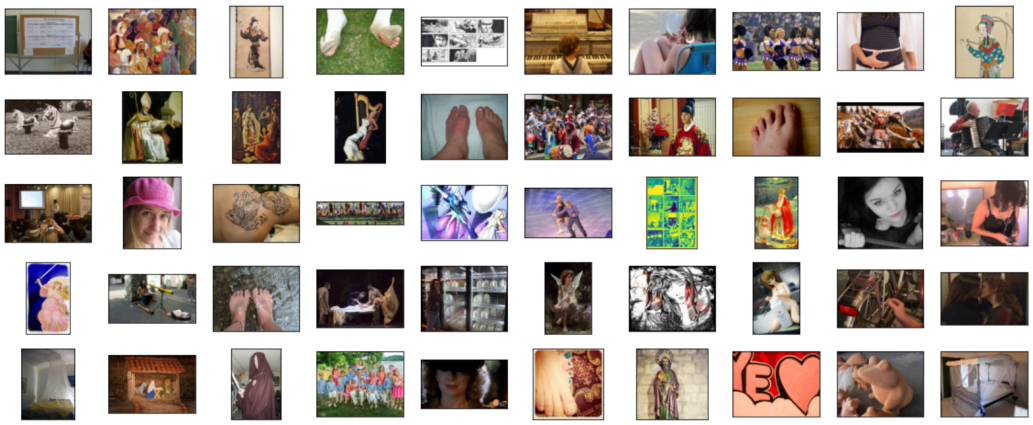

Figure 51: Mammal+woman

B Open Images Analysis

B.0.1 Pretrained Dataset: Open Images

In Figure 52, we observe that the trends for ImageNet21K and the 400M images from the web are relatively similar whereas the ImageNet1K trends are different. This shows that ImageNet 21K and the 400M images from the web have a similar effect on the biaess in the pretrained feature representation space.

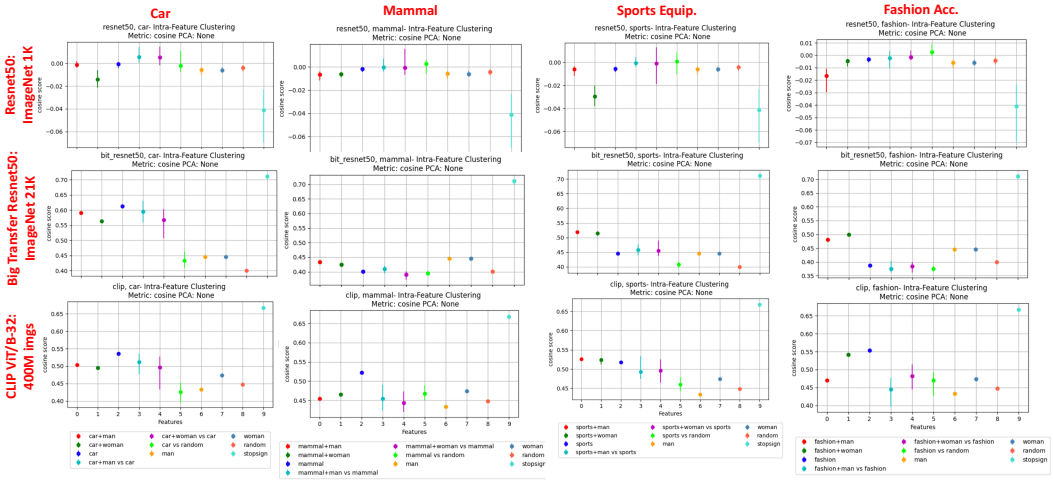

Figure 52: ResNet50, Big Transfer ResNet50, and CLIP ViT/32-B: intra-class variation and embedding distance for classes of interest in the Open Images analysis set: Car, Mammal, Sports, Fashion. A similar trend across CLIP ViT/32-B and Big Transfer 21K implies that both of these datasets encode and reflect similar biases in the pretrained space.

In Figure 53, we observe that the Big Transfer ResNet50 model was almost unaffected by finetuning on the Open Images dataset whereas the ResNet50 model was greatly impacted since the trends between the pretraining and finetuning results are very different.

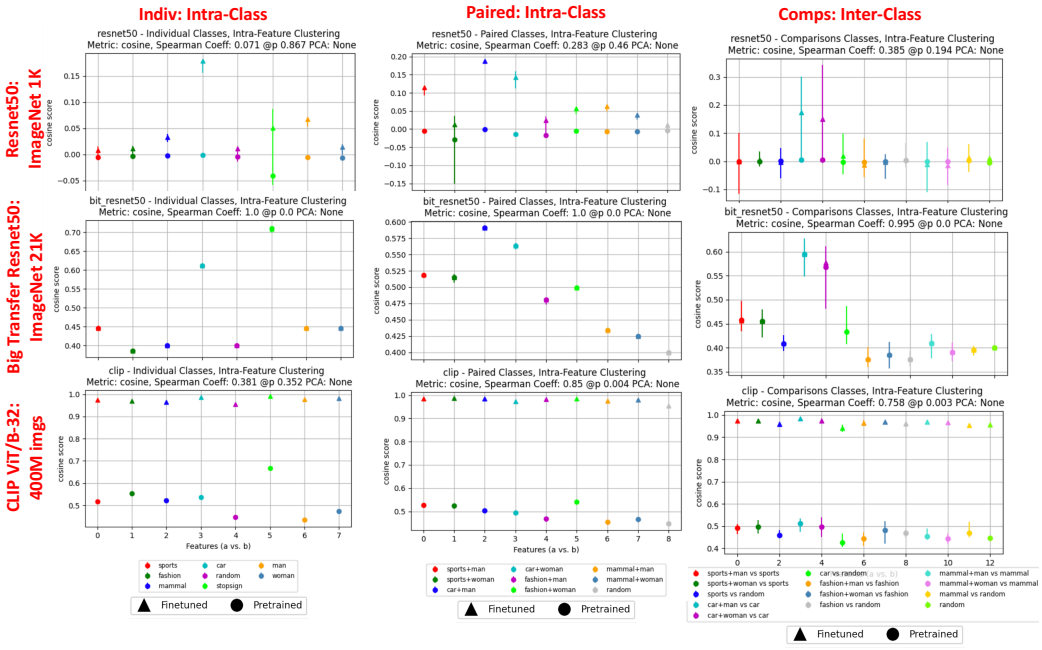

Figure 53: ResNet50, Big Transfer ResNet50, and CLIP ViT/32-B: these plots segment the categories in the Open Images v4 analysis set by individual classes (car, refrigerator, surfboard, etc.), paired classes (refrig+man, surfb+woman etc.), and comparisons (refrig vs. random). These plots show the results of the intra-class variation metric and the embedding distance metric of each class before and after finetuning.

We can more closely observe this in Figure 54, where we examine the trends between the pretrained and finetuned scores for each of the models. Big Transfer ResNet50's trend is almost linear which matches what we observed in Figure 53. However, the trends for Clip ViT/B-32 and ResNet50 are not linear implying that finetuning greatly impacted the model's biases and the biases from the pretrained model were not preserved. So we observe that models pretrained on ImageNet21K are less prone to changes in their biases after finetuning on Open Images, whereas pretraining on ImageNet1K and 400M images from the web are more prone to absorbing and reflecting biases from Open Images.

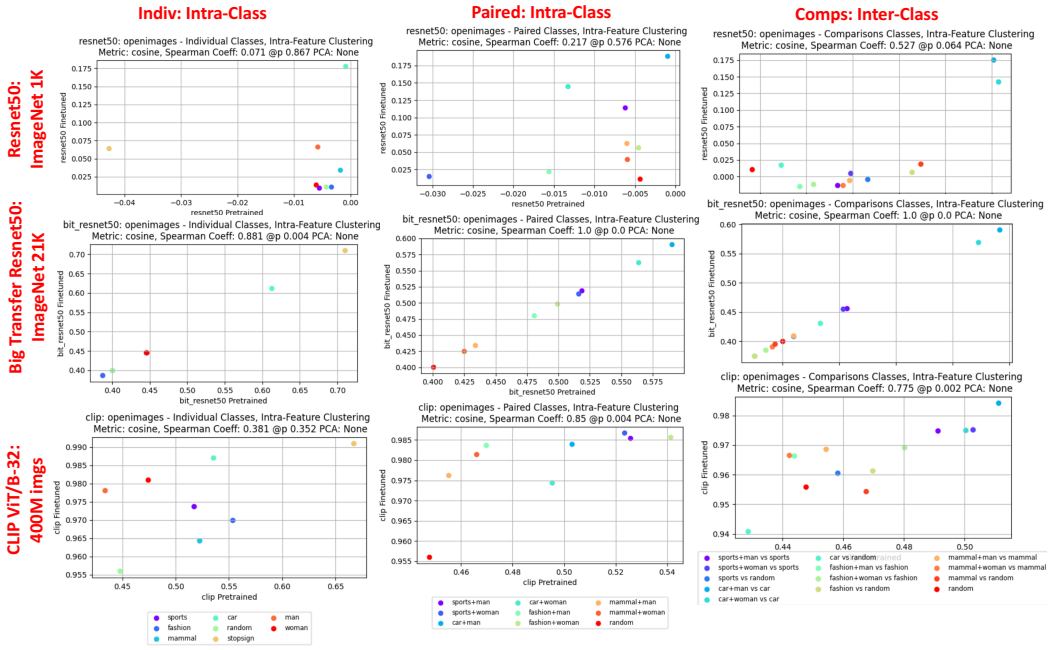

Figure 54: ResNet50, Big Transfer ResNet50, and CLIP ViT/32-B: Plots the trend between the pretrarining and finetuning scores. A linear trend for Big Transfer ResNet50 shows that this model preserved its biases from pretraining to fineetuning whereas the less linear trends in ResNet50 ND CLIP ViT 32-B implies that biases from finetuning impacted the feature representations in the finetuned feature space.

B.0.2 Network Architecture: Open Images

Figure 55 shows results from ResNet18, ResNet50, and CLIP ViT/B-32. We observe a similar result to the COCO2017 dataset where ResNet18 and ResNet50 exhibit similar trends implying that despite being different architectures, their biases are still the same because the dataset they were pretrained on is the same. However, for the CLIP architecture, we cannot necessarily discern whether the biases are coming from the pretraining dataset or the network architecture. But nonetheless, the biases are different from ResNet18 and ResNet50.

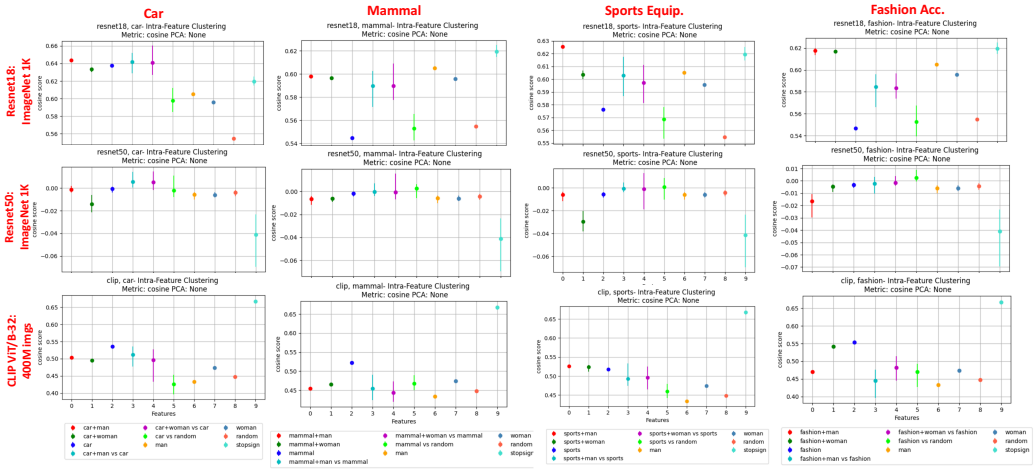

Figure 55: ResNet18, ResNet50, CLIP ViT/B-32: Analysis of biases in the pretrained feature space of categories in the Open Images analysis set.

Figures 56 and 57 show the impacts of finetuning on Open Images. We observe that ResNet18 preserved most of its biases from its pretrained model since the trend between the pretrained and finetuned scores is close to linear. However, this is not the case for ResNet50. This could indicate that the difference in network architecture could impact the biases in the finetuned feature space when finetuning on Open Images. This was not true for the COCO dataset as observed in Figure 10.

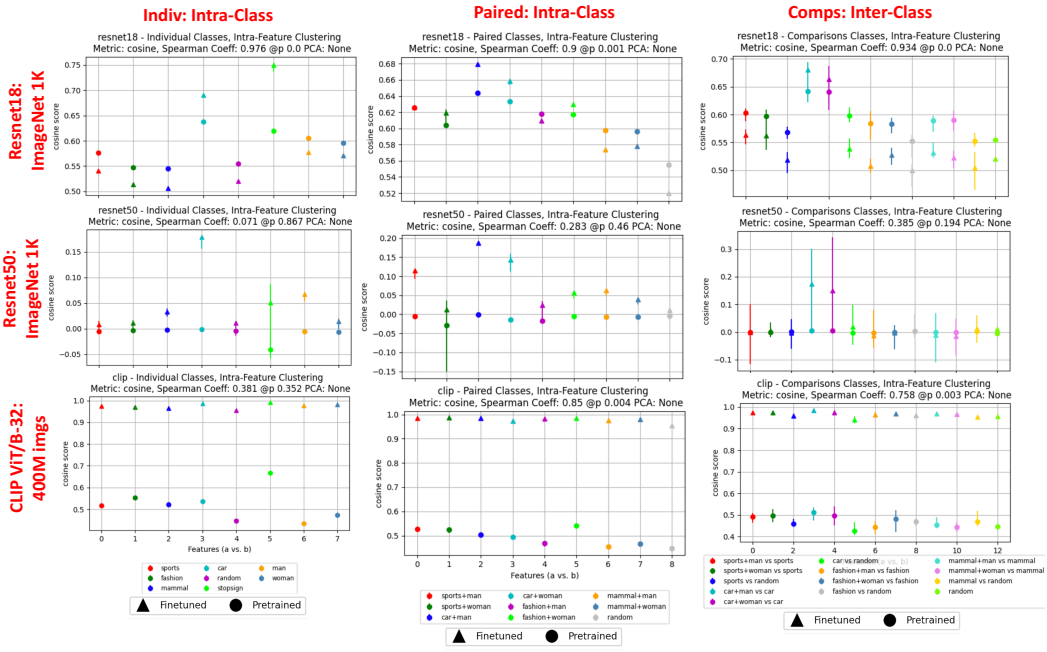

Figure 56: ResNet18, ResNet50, CLIP ViT/B-32: Analysis of biases before and after finetuning on the Open Images analysis set.

Figure 57: ResNet18, ResNet50, CLIP ViT/B-32: Trends between pretraining and finetuning scores.

B.0.3 Training Setting: Open Images

Figure 58 shows that the trends in biases are very different between ResNet50 and MoCo ResNet50 implying that the pretraining setting does have an impact on how the biases are represented in the pretrained features. f

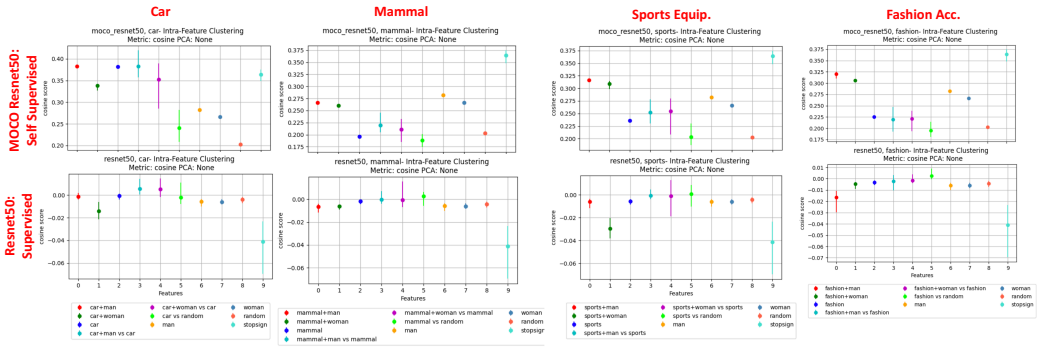

Figure 58: ResNet50, MoCo ResNet50: Analysis of biases in the pretrained feature space of categories in the Open Images analysis set.

From Figure 59 and 60, we find that MoCo ResNet50 preserves most of its biases from pretraining whereas ResNet50's biases are very different. This could implying that models pretrained in supervised setting are more prone to absorbing biases in the dataset they were finetuned on than models pretrained in a self supervised setting.

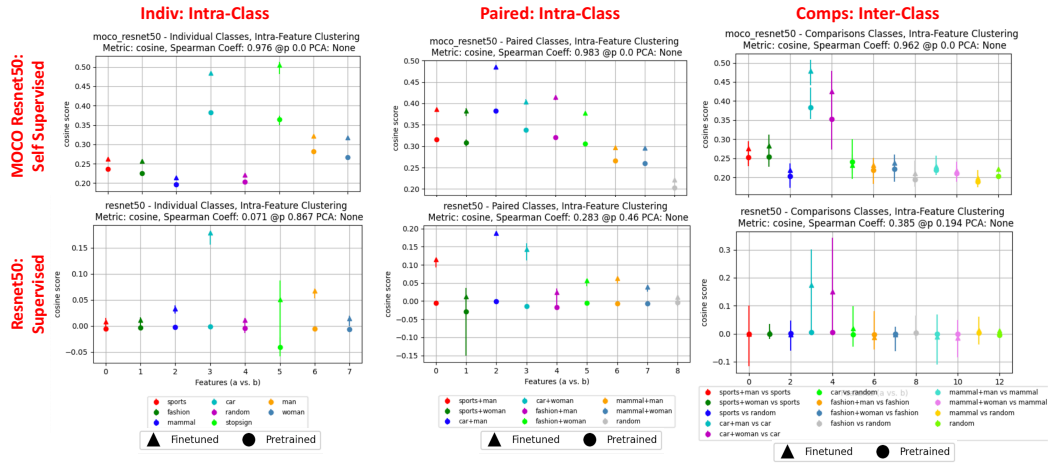

Figure 59: ResNet50, MoCo ResNet50: Analysis of biases before and after finetuning on the Open Images datset.

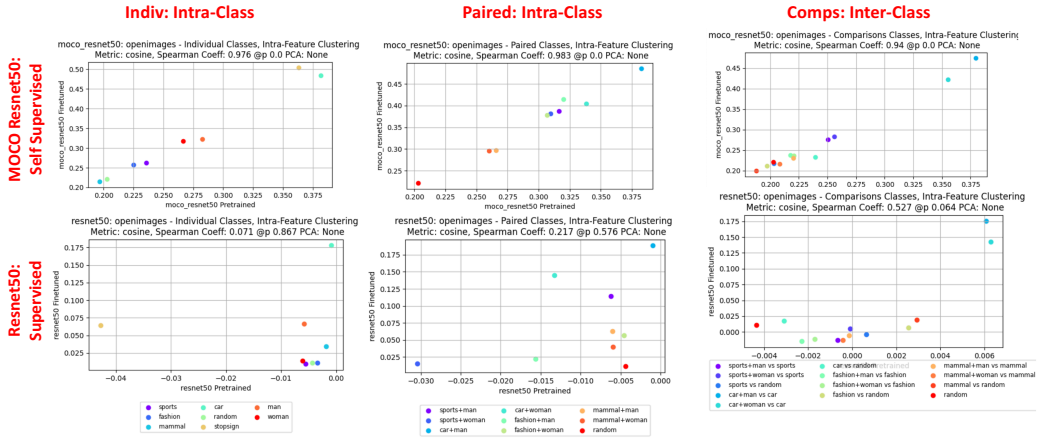

Figure 60: ResNet50, MoCo ResNet50: Trends between pretraining and finetuning scores.

B.0.4 Model Comparison: Open Images

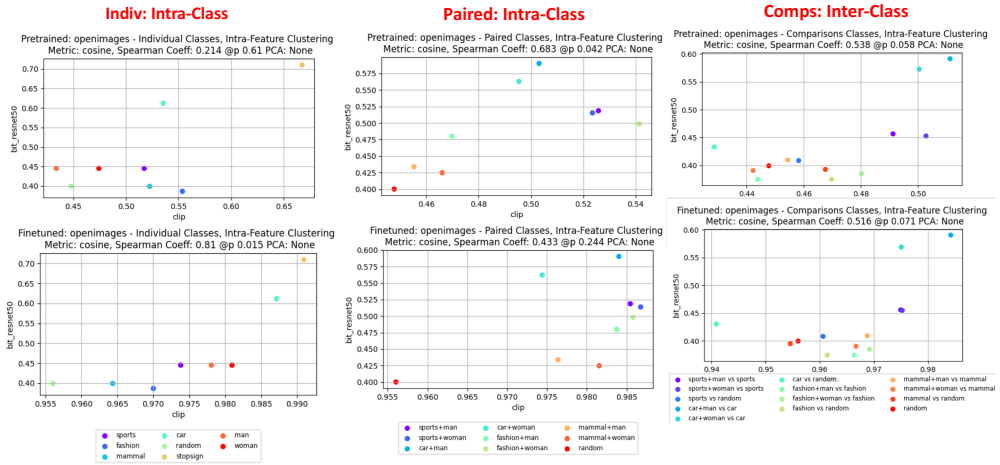

Figure 61: Comparing CLIP ViT/B-32 and Big Transfer. We find that the trends for the individual class plots are more linear after finetuning implying that the models absorbed biases from the Open Images dataset for these specific classes. However, this is not true for the paired and comparisons plots. Because the trend before finetuning is not linear either, it is difficult to discern whether the models were impacted by biases after finetuning from these plots and we wouuld need to examine each of these biases more carefully.

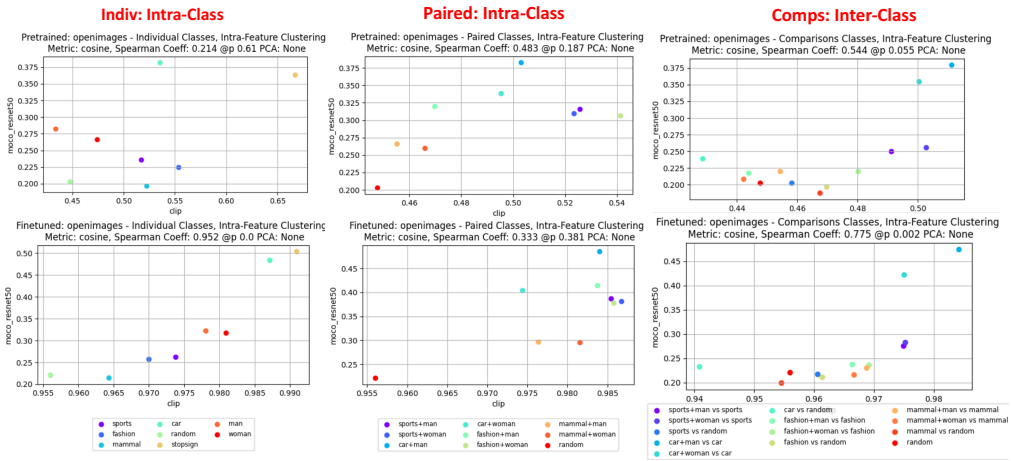

Figure 62: Comparing CLIP ViT/B-32 and MoCo ResNet50. Similar to Figure 29, we observe that the trends for the individual class plots and the comparison class plots become more linear after finetuning so we can reasonbly assume that both of these models encoded biases from the Open Images after finetuning for those specific classes.

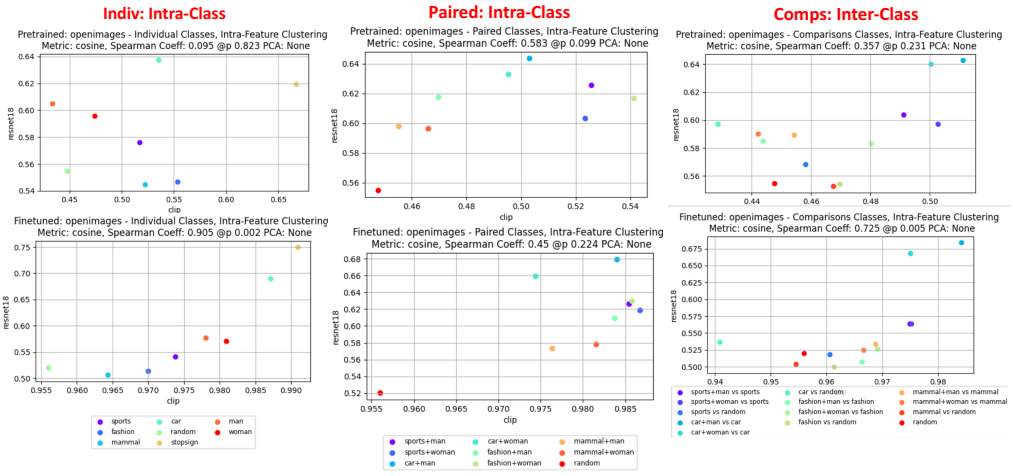

Figure 63: Comparing CLIP ViT/B-32 and ResNet18. Each of the plots shows a higher linear trend after finetuning implying that both of these models absorbed and reflected biases from the Open Images dataset. However, we cannot conclude which model preserved more of their biases from their pretrained weights without observing the biases of these models independently.

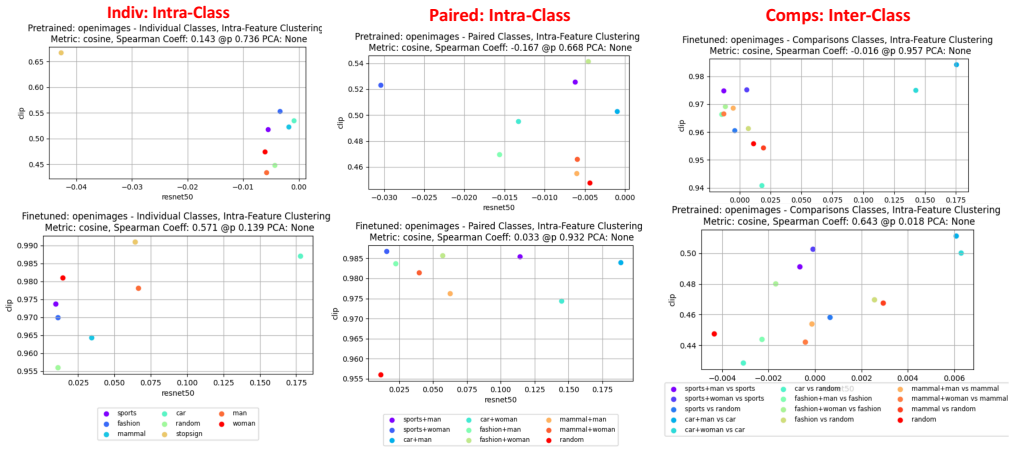

Figure 64: Comparing CLIP ViT/B-32 and ResNet50. Because the trends before and after finetuning are not linear, we cannot conclude much about the biases of these models except that they represent their biases very differently in the pretrained and finetuned feature space.

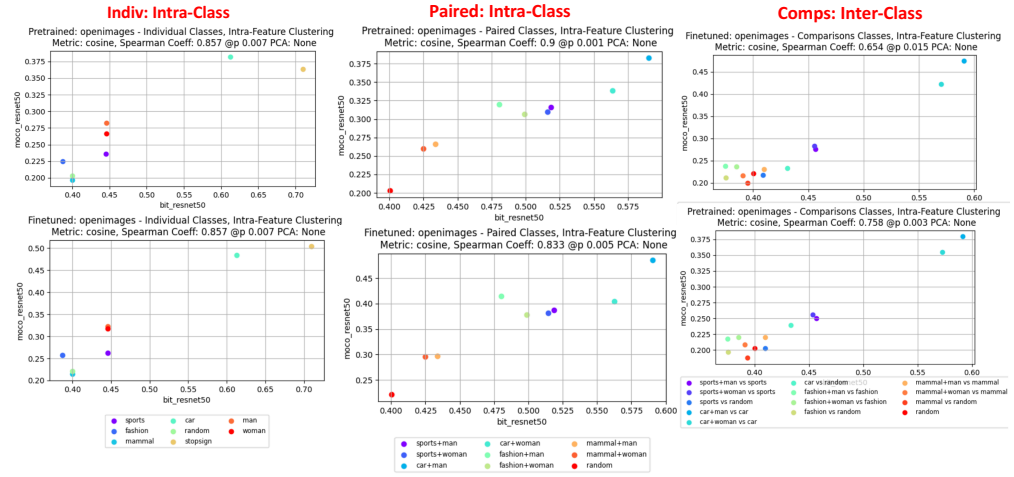

Figure 65: Comparing Big Transfer ResNet50 and ResNet50, the trends before and after finetuning are mostly linear implying that these models encode their biases in the pretrained and finetuned feature space. However, we cannot conclude if both of these models were impacted by biases in the finetuning dataset since the pretraining trend is also linear. There is the possiblity that both of these models preserved their biases from their pretrained weights.

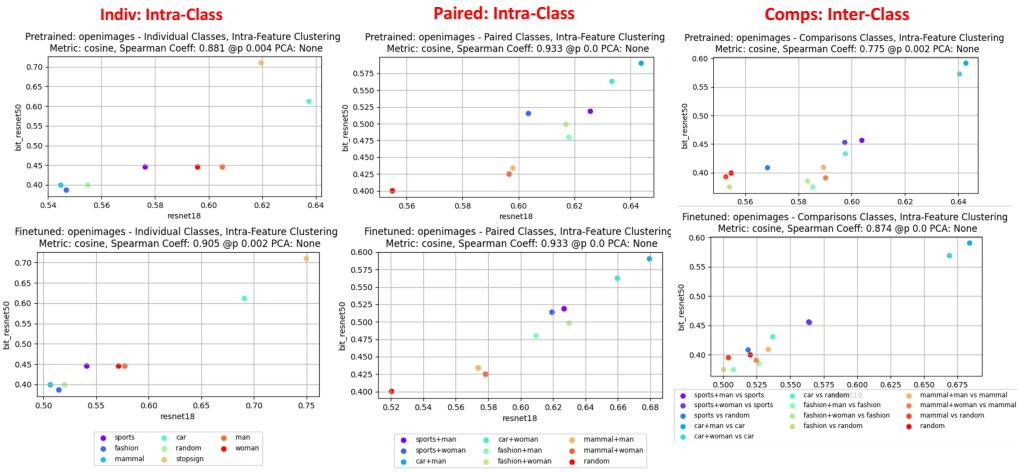

Figure 66: Comparing ResNet18 and Big Transfer ResNet50. Similar to Figure 33, both the pretrainig and finetuning trends are similar and close to linear, as a result, we cannot conclude whether the biases came from the finetuning dataset. We can only conclude that the biases for these two models were represented in a similar way.

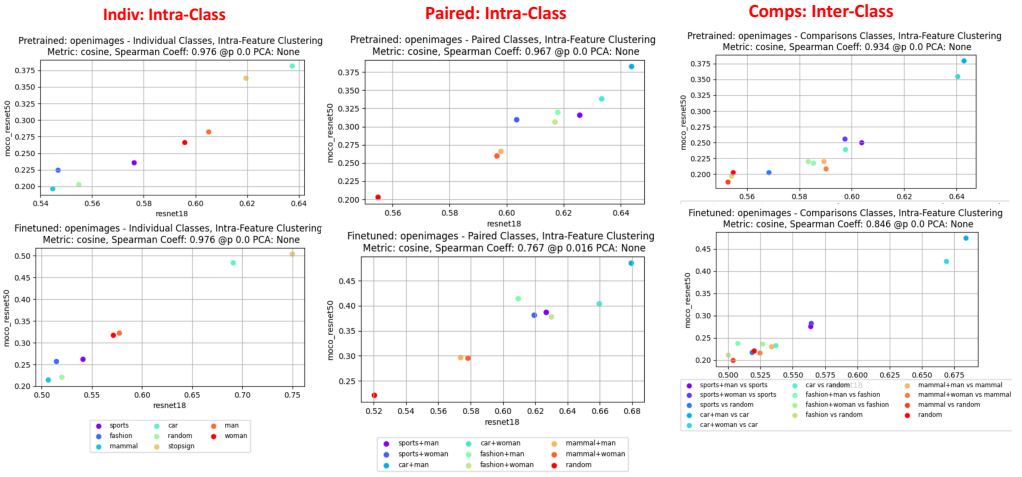

Figure 67: Comparing ResNet18 and MoCo Resnset50, similar to Figures 33 and 34, we cannot conclude whether the biases came from the finetuning dataset. We can only conclude that the biases for these two models were represented in a similar way.

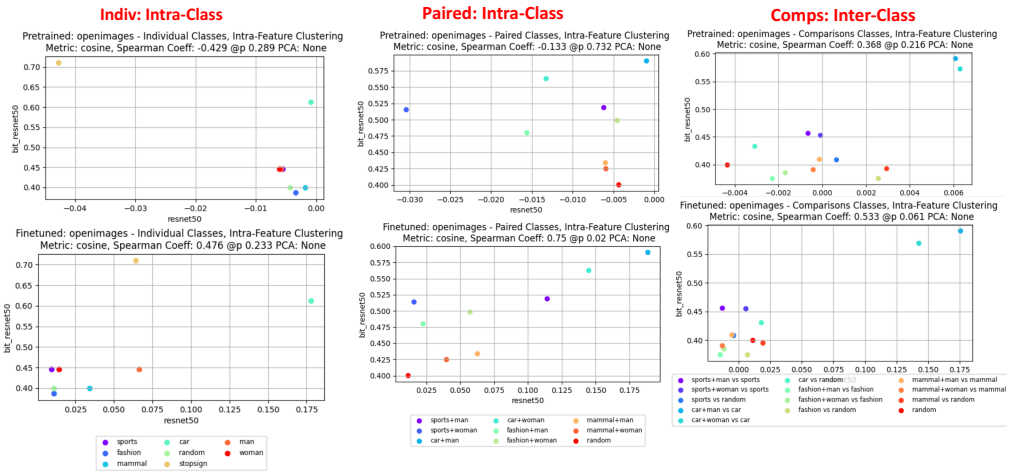

Figure 68: Comparing ResNet50 and Big Transfer ResNet50, The biases after finetuning are more linear than before finetuning. This implies that the biases that the models encoded before finetuning were very different. Since we are comparing Big Transfer ResNet50 and ResNet50, we can attribute these biases to the dataset they were pretrained on. However, they are more linear after finetuning implying that they both absorbed some biases from finetuning on Open Images.

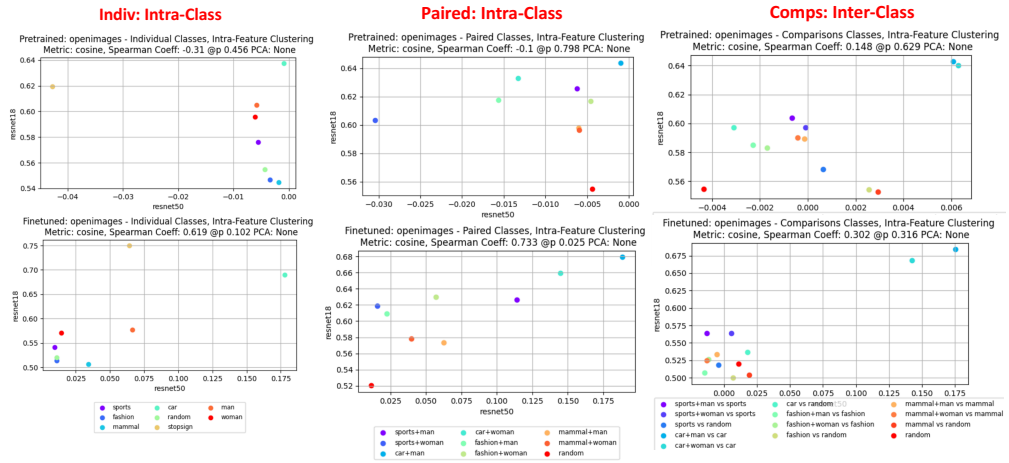

Figure 69: Comparing ResNet50 and ResNet18. Similar to Figure 36, The trends after finetuning become more linear implying that the models encoded their pretrained biases very differently due to their differing architectures, but absorbed some biases from Open Images after finetuning.

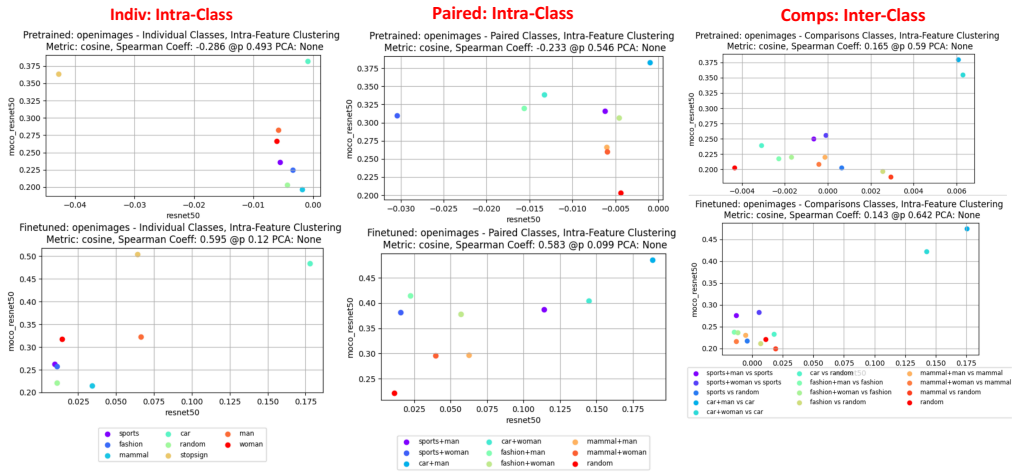

Figure 70: Comparing ResNet50 and MoCo ResNet50, we can make the same conclusions as Figure 37. The trends after finetuning become more linear implying that the models encoded their pretrained biases very differently due to being pretrained on different datasets, but absorbed some biases from Open Images after finetuning.

References

[1] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021.

[2] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[3] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander J. Smola. Resnest: Split-attention networks. *CoRR*, abs/2004.08955, 2020.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[5] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[9] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[10] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *CoRR*, abs/1707.09457, 2017.

[11] Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. *CoRR*, abs/2106.08503, 2021.

[12] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *CoRR*, abs/2108.08810, 2021.

[13] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *CoRR*, abs/1607.06520, 2016.

[14] *Spearman Rank Correlation Coefficient*, pages 502–505. Springer New York, New York, NY, 2008.

[15] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, and et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, Mar 2020.

[16] Dora Zhao, Angelina Wang, and Olga Russakovsky. Understanding and evaluating racial biases in image captioning. In *International Conference on Computer Vision (ICCV)*, 2021.

[17] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528, 2011.

[18] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.

[19] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. Gender bias in contextualized word embeddings. *CoRR*, abs/1904.03310, 2019.

[20] Yuchi Tian, Ziyuan Zhong, Vicente Ordonez, and Baishakhi Ray. Testing deep neural network based image classifiers. *CoRR*, abs/1905.07831, 2019.

[21] Ignacio Serna, Alejandro Peña, Aythami Morales, and Julian Fiérrez. Insidebias: Measuring bias in deep networks and application to face gender biometrics. *CoRR*, abs/2004.06592, 2020.

[22] Ryan Steed and Aylin Caliskan. Image representations learned with unsupervised pre-training contain human-like biases. *CoRR*, abs/2010.15052, 2020.

[23] Zhiheng Li and Chenliang Xu. Discover the unknown biased attribute of an image classifier. *CoRR*, abs/2104.14556, 2021.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[25] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Large scale learning of general visual representations for transfer. *CoRR*, abs/1912.11370, 2019.

[26] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.