A Conservative Approach for Unbiased Learning on Unknown Biases

Myeongho Jeon[1], Daekyung Kim[2], Woochul Lee[1], Myungjoo Kang[1], Joonseok Lee[1]
[1]Seoul National University, [2]Monitor Corporation
{andyjeon, sunnmoon137, woochulee, mkang, joonseok}@snu.ac.kr

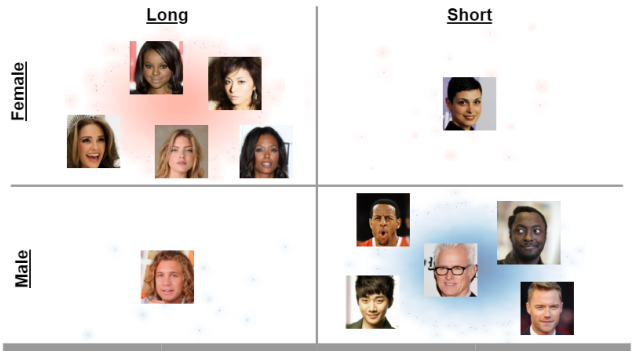
Abstract

Although convolutional neural networks (CNNs) achieve state-of-the-art in image classification, recent works address their unreliable predictions due to their excessive dependence on biased training data. Existing unbiased modeling postulates that the bias in the dataset is obvious to know, but it is actually unsuited for image datasets including countless sensory attributes. To mitigate this issue, we present a new scenario that does not necessitate a predefined bias. Under the observation that CNNs do have multi-variant and unbiased representations in the model, we propose a conservative framework that employs this internal information for unbiased learning. Specifically, this mechanism is implemented via hierarchical features captured along the multiple layers and orthogonal regularization. Extensive evaluations on public benchmarks demonstrate our method is effective for unbiased learning.[1]

1. Introduction

Recently, machine learning models (*e.g.*, convolutional neural networks) achieve state-of-the-art performance on image classification. However, the model could be often biased, as it is trained overly dependent on the distribution of the training dataset [17, 20, 21]. The biased model is vulnerable to unreliable generalization to unseen data. For instance, suppose we classify the gender of a person in an image. A naturally collected image dataset often contains significantly more examples with (female, long hair) and (male, short hair) than the other combinations, as seen in Fig. 1. Although the hair length is not biologically correlated to gender, the model is subject to be confused as if it is, due to the high correlation observed in the data. In this case, we call that this gender classification dataset is highly *biased* in hair length.

For better generalization, it would be important to train the classifier unbiased, and a line of recent works is dedicated to this problem. The easiest problem setting is the *labeled bias*, where we know what the bias is and each training example is annotated not only with the actual label but also with the bias. For example, we know the gender classification dataset is biased in hair length and each image is annotated with the hair length as well as the gender, corresponding to the row *Labeled Bias* in Fig. 1. As the bias can be easily quantified, several supervised learning approaches [1, 2, 6, 8, 12] were proposed to unlearn the bias.



Bias Category	Label	Bias	Bias Label
Labeled Bias	Male / Female	Hair	Long / Short
Known Bias	Male / Female	Hair	?
Unknown Bias	Male / Female	?	?

Figure 1. **Categorization of Data Bias Problems** with an example of gender prediction biased in hair length. The most limited case, *unknown bias*, is what we address in this paper.

However, not all classification datasets provide annotations for the bias variable even if we know what it is. For instance, it is more common that we know the gender classification dataset may be biased in hair length, but each individual is annotated only with the gender, not with the hair length. We call this *known bias* as in Fig. 1. It is more challenging than *labeled bias*, since it is hard to unlearn the bias by training a supervised model directly taking advantage of the annotations. A style-transferred image dataset [7] and an adversarial training against structurally limited network [3] were proposed to address *known bias* in texture. A traditional image processing algorithm was employed as bias-robust filtering in the deep neural network [22].

[1]Source code: https://github.com/aandyjeon/UBNet

Although the listed methods above are robust on the target (known) biases, they are limited in unlearning other unexpected biases. In fact, a collection of images is subject to dozens of biases, whether strongly or not, and thus the known bias may not be the only one to take into account. For example, the gender classification with hair length as its known bias can be also biased in age, skin tone, lighting condition, and more. The last setting in Fig. 1, namely *unknown bias*, formulates such a condition, where we do not know even what is a notable bias in the dataset. As this setting gives no information about the bias, the aforementioned existing methods are not applicable.

To tackle this *unknown bias* condition, we aim to design a conservative learning framework. As its first step, we conduct a motivating experiment to understand how biased representations are obtained. From a previous study [7], CNN models were observed to be over-confident in a certain feature. We hypothesize that this tendency induces the classifier less robust on the biased conditions. That is, the model uses the biased feature as a strong candidate for learning, and thus its dependence on the bias gets higher than that of the other features. Utilizing the feature maps captured inside the CNN, we estimate the degree of bias of feature maps in the model, using a tool called norm activation [18] (Sec. 3). From this experiment, we find out that the features immediately before the prediction layers are significantly more biased than lower-level features.

Based on these observations, we design a novel framework, namely *unbiasing network (UBNet)*, that exploits hierarchical features and orthogonal regularization. Instead of classifying based only on the top-most layer of the network, our framework conservatively employs *hierarchical features* (Sec. 4.1) from multiple levels of the network, widening referred feature space as well as utilizing less biased representations. Also, *orthogonal regularization* [23] (Sec. 4.2) is applied in the encoding phase to encourage independence between hierarchical features. Extensive evaluation on multiple biased datasets exhibits our proposed framework is effective in unbiased learning. Our main contributions are summarized as follows:

- We present a new framework that addresses *unknown bias*, where no specific information about the bias is provided.
- We propose a novel unbiased learning method with *hierarchical features* and *orthogonal regularization*, designed to conservatively employ the features already captured by the base model with only a few additional parameters.
- Extensive experiments exhibit that UBNet contributes to the model's robustness. Especially, it is remarkable to note that our UBNet generalizes even better than the competing models in the *known bias* setting (texture), significantly disadvantageous for our model.

2. Problem Statement

Unbiased modeling. Consider a binary classification problem from a multivariate instance (*e.g.*, image) $\mathbf{x}$ to a label $\{0, 1\}$. For a collection $\mathcal{X} = \{\mathbf{x}_i : i = 1, ..., N\}$ with N instances, consider a label space $\mathcal{Y}$, a set of all possible assignments of 0 or 1 to each instance in $\mathcal{X}$. A label $\mathbf{y} \in \{0, 1\}^N$ is a particular way of assigning either 0 or 1 to each instance. Some $\mathbf{y} \in \mathcal{Y}$ may convey physical meaning, potentially ranging from fine details (*e.g.*, the pixel at (0, 0) is black) to high-level semantics of the image (*e.g.*, there is a horse). Binary classification is a task to learn the mapping from $\mathcal{X}$ to a particular $\mathbf{y}_{\text{target}} \in \mathcal{Y}$, maximizing generalizability to unseen examples sampled from the same distribution with $\mathcal{X}$.

Now, consider another $\mathbf{y}' \in \mathcal{Y}$ which is $\mathbf{y}' \neq \mathbf{y}_{\text{target}}$. Most $\mathbf{y}'$ may be usually independent of the $\mathbf{y}_{\text{target}}$, but some might have significant but spurious correlation with the target. We call that the dataset $\mathcal{X}$ is *biased* in the variable $\mathbf{y}'$, as $\mathbf{y}'$ potentially can mislead the model at inference. In a face dataset, for instance, the label $\mathbf{y}_{\text{gender}} = \{\text{female}(0), \text{male}(1)\}$ or $\mathbf{y}_{\text{hair}} = \{\text{long}(0), \text{short}(1)\}$ may be highly correlated unless the dataset is specially designed. Thus, when one of them is the target label, the dataset can be easily biased in the other.

If we train a classifier f on a biased dataset, f may perceive the biased labels as important information for target prediction. As a consequence, f has a tendency to lean towards the biased labels; for example, it may bet the long hair on female and short hair on male. Therefore, the goal of the unbiased modeling is to learn unbiased representations or reduce model's feature dependence on biased labels, while maintaining the target label prediction performance.

One might argue that unbiased modeling and domain adaptation tackle the same problem, as both of them aim at better generalizing to unseen test examples from a different distribution. However, they are indeed different problems [3]. Specifically, domain adaptation is a task to generalize from a training set with skewed distribution on some variable $\mathbf{y}'$ (but still it is independent of $\mathbf{y}_{\text{target}}$) to its opposite. For instance, a gender classification model is trained on a dataset with seniors only and evaluated for young people. The unbiased learning problem, on the other hand, tackles a circumstance that one or more variables have a strong correlation to the target variable $\mathbf{y}_{\text{target}}$. For example, the gender classification model is trained mostly on female + long hair and male + short hair combinations, then is expected to correctly classify the gender of females with short hair or males with long hair.

Our problem setting. We address the *unknown bias* in Fig. 1, where all about bias is opaque. That is, we do not know which $\mathbf{y}' \in \mathcal{Y}$ is highly correlated to $\mathbf{y}_{\text{target}} \in \mathcal{Y}$ in the dataset.

One might claim it is impossible to completely debias the model from *all* unknown biases, since we do not know what they are, unlike a known bias which we can explicitly model not to rely on it. What we intend with the proposed approach is making the model more *robust* and *conservative*, leading it to rely on more variety of cues, preventing it from over-committing to a specific cue. This is particularly desirable in practice, where a model is deployed for years and the query distribution changes. In this sense, under the definition of "bias" as a misleading feature, we mean by "unbiasing" regularizing the model from being over-dependent on any particular feature, instead of making it independent of any cue. In other words, we mean the opposite: making the model rely on more cues to be less biased.

3. Motivation

In this section, we first investigate how a standard CNN model learns biased representations. We detail our observation that CNN models tend to learn a few specific aspects from the training set, as opposed to learning a variety of features (in Sec. 3.1), and that their higher level features close to the classification head tend to be biased more (in Sec. 3.2). Combining with the fact that a standard CNN solely relies on the very last layer to classify, we are going to motivate to use multi-variant features learned from the entire hierarchy of the CNN model in the next section.

3.1. Feature Dependence in CNN

Geirhos *et al*. [7] investigated the contributions of various attributes on the predictions by a standard CNN model pretrained on ImageNet. They distorted original test images to intentionally rely on a specific attribute (corresponding to a $\mathbf{y}' \in \mathcal{Y}$), as seen in Fig. 2. Then, they estimated the model's performance on each distorted sample. Despite never being directly trained on any of these deformed images, the model was able to classify texture images as accurately as the original images, while it failed to classify images with other types of distortion.

We interpret this as over-credence towards a specific feature. In other words, relying rarely on other features makes the model more vulnerable to biased learning. That is, if the correlation between the texture (bias) and target label $\mathbf{y}_{\text{target}}$ is subtle at testing, the biased model loses the rationale for the discrimination. This observation hints us to mitigate imbalanced reference on certain dominating features in order to make the model more robust to bias.

3.2. Measuring Bias in CNN

We further investigate how a multilayered convolutional network learns biased representations in detail. Specifically, we analyze the degree of bias in each layer using Norm activation [18] as a tool.

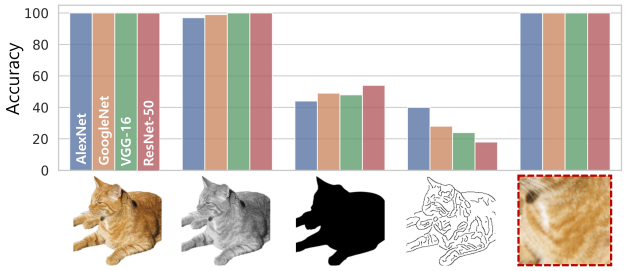


Figure 2. **Feature dependence of the CNNs.** The accuracy on original, grey-scaled, silhouette, edges, and texture are estimated for ImageNet-pretrained CNN models. (from the Fig. 2 in [7])

Norm activation. Norm activation [18] estimates the robustness of a layer in a deep network on the dataset. Hence, the difference between activation norms on two reversely biased datasets in a specific feature represents how much the certain layer in the model is dependent on that bias. For an activation map $\mathcal{A}_c^{[l]}$ of the channel c at layer l, its spatially averaged activation map is denoted as $\overline{\mathcal{A}}_c^{[l]} \in \mathbb{R}$. We denote its maximum across all channels as $\lambda^{[l]} \in \mathbb{R}$:

$$\lambda^{[l]} = \max_c (\overline{\mathcal{A}}_c^{[l]}), \tag{1}$$

intuitively indicating the degree of maximal activation at layer l. We compute $\lambda^{[l]}$ at all layers in the network, and normalize across them:

$$\lambda'^{[l]} = \frac{\lambda^{[l]}}{\max_l \lambda^{[l]}}. \tag{2}$$

Now, suppose we train the exactly same model k times under the same condition except for weight initialization. The *norm activation* $r_d^{[l]}$ for a biased test data d is defined as

$$r_d^{[l]} = \frac{\min_i \lambda'^{[l]}(f_d^{(i)})}{\max_i \lambda'^{[l]}(f_d^{(i)})}, \tag{3}$$

where $f^{(i)}$ denotes the i^{th} model, wth $i = 1, ..., k$. Intuitively, the $r_d^{[l]}$ indicates how much the model's decision changes for d across k learned models. $r_d^{[l]} \approx 1$ if the model is robust, while $r_d^{[l]}$ gets smaller towards 0 otherwise.

How much each layer is biased. We investigate two standard CNN models, VGG11 [19] and ResNet18 [9]. We sample l whenever the size of the feature maps are halved, and set $k = 5$. For gender prediction, we design two test sets: $d_1 = \{(\text{female, long hair}), (\text{male, short hair})\}$ and $d_2 = \{(\text{female, short hair}), (\text{male, long hair})\}$. The models are trained on training data close to d_1.

Figure 3 illustrates $\Delta r^{[l]} \equiv r_{d_1}^{[l]} - r_{d_2}^{[l]}$, the difference of norm activations on the two test sets. The larger the gap is, the less the model is robust on the biased test set d_2, indicating the layer's activation is more biased in hair length.

Figure 3 shows the last layer is extensively biased compared to the other layers. This experiment exhibits that the use of features from lower layers potentially encourages the model to exploit less biased representations.

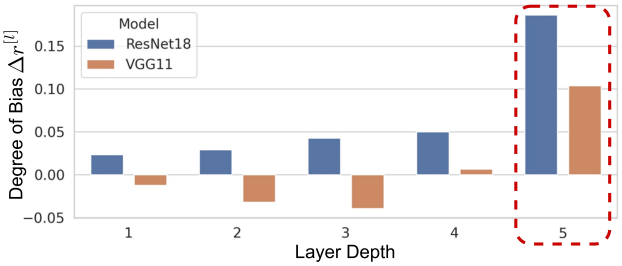


Figure 3. **Degree of Bias.** The y-axis measures degree of bias by $\Delta r^{[l]}$, the difference of norm activations on two test sets.

4. UBNet: The Proposed Method

In this section, we present the proposed model *Unbiasing Network (UBNet)*, illustrated in Fig. 4. Particularly, the *hierarchical features* and *orthogonal regularization* are the operative ideas to assist our motivation in Sec. 3. These two strategies are applied to a CNN-based model, which is referred to as the 'base model' henceforth. Then, hierarchical features learned by the base model are concatenated and fed into an orthogonally regularized classifier, called *Ortho-Block*, for discrimination. This framework can be applied to an arbitrary CNN with minimal overhead, as it adds only a few parameters (*e.g.*, in our experiment, 0.21% of parameters are added over the base model).

4.1. Hierarchical Features

Most standard CNN models classify an image purely based on the activation from the top-most layer. However, we illustrate in the previous section that 1) the CNN's over-credence on certain features causes a biased classifier (Sec. 3.1) and 2) the last layer in the conventional CNN-based image classifier is much biased than the lower-level layers (Sec. 3.2). Thus, the standard CNN models relying solely on their highest-level representation are particularly vulnerable to being biased.

For unbiased inference, we exploit *hierarchical features* captured inside the CNN. A CNN-based image classifier extracts hierarchical features from simple patterns (*e.g.*, corners or edge/color conjunctions) in its lower layer to more complex high-level features (*e.g.*, significant variation and class-specific features) in its higher layer because of the spatially limited convolution operation [24]. As all these features can be considered as more multifarious features than only the high-level features, the hierarchical features represent multi-variant features. Hierarchical features also represent less biased features, since the lower-level layers contain less skewed features as seen in the motivating experiment in Sec. 3.2.

The set of hierarchical feature maps $\mathcal{H}$ of a CNN with L layers can be expressed as $\mathcal{H} = \{\mathbf{h}_1, \dots, \mathbf{h}_L\}$, where $\mathbf{h}_l \in \mathbb{R}^{W_l \times H_l \times C_l}$ denotes the feature maps at layer l of size $W_l \times H_l$ with C_l channels, sequentially extracted from each layer in the base model. To enable concatenation of $\mathbf{h}_l$ of different shapes for $l = 1, ..., L$, f_{trans} first converts the input feature map size at each layer to $W \times H \times C$. Specifically, it applies an average pooling to shrink the feature map to $W \times H$ and 1×1 convolution to adjust (either increase or decrease) the number of channels to C. Here, W, H, and C are hyperparameters, such that $W \leqq W_l$ and $H \leqq H_l$ for all $l = 1, ..., L$. The resulting feature maps

$$\mathbf{g}_l = f_{\text{trans}}(\mathbf{h}_l) \in \mathbb{R}^{W \times H \times C}, \quad \forall l = 1, \dots, L \tag{4}$$

are then concatenated to $\mathcal{G} = [\mathbf{g}_1, \dots, \mathbf{g}_L] \in \mathbb{R}^{L \times W \times H \times C}$, where $[,]$ denotes concatenation. We term each $\mathbf{g}_l$ as a *group feature* at layer l, capturing semantics at different level (low to high) since they are originated from different layers.

4.2. Orthogonal Regularization

Group Convolution. In spite of providing the hierarchical semantic information, the high-level features might still dominate the others if the group features from multiple levels are freely fused, either by fully-connected or convolutional layers, causing the encoded representations to be biased again following the same way of the base model. For this reason, the multivariate features need to be treated independently on each group before making final prediction. Instead of fully-connected or regular convolution, we resort to use group convolution [10], where all the feature maps in the same group $\mathbf{g}_l \in \mathcal{G}$ are weight-connected but separated from the other group features. In other words, the interrelation between hierarchical features is restricted, preserving distinct features respectively. Group convolution $f_g(\mathcal{G})$ performs convolution operations on each group $\mathbf{g}_l \in \mathcal{G}$ independently, producing $\mathcal{S} = [\mathbf{s}_1, \dots, \mathbf{s}_L] \in \mathbb{R}^{L \times W \times H \times C}$.

Afterwards, we apply channel-wise spatial fusion, again to avoid indiscreet fusion between group features. As in the Ortho-Trans box in Fig. 4, each channel $\mathbf{s} \in \mathbb{R}^{L \times W \times H}$ in $\mathcal{S}$ is mapped to a scalar by a linear layer f_s with weight $\mathbf{W}^s$. The output from f_s is denoted as $\mathcal{O} \in \mathbb{R}^{L \times 1 \times 1 \times C}$; that is, $\mathcal{O} = [o_1, \dots, o_L] = f_s(\mathcal{S})$.

Orthogonality. In addition, we apply *orthogonal regularization* [23] to encourage the independence between group features. We define the group convolution layer and the spatial weighting layer in conjunction with orthogonal regularization as *Ortho-Conv* and *Ortho-Trans*, respectively, as shown in Fig. 4. We term the combination of these two layers as *Ortho-Block*. The Ortho-Block f_o takes $\mathcal{G}$ as input and outputs activated values as where each $\mathbf{o}_l \in \mathbb{R}^{1 \times 1 \times C}$ is made from each $\mathbf{g}_l$. When the convolutional filters and spatial weights of different group features are

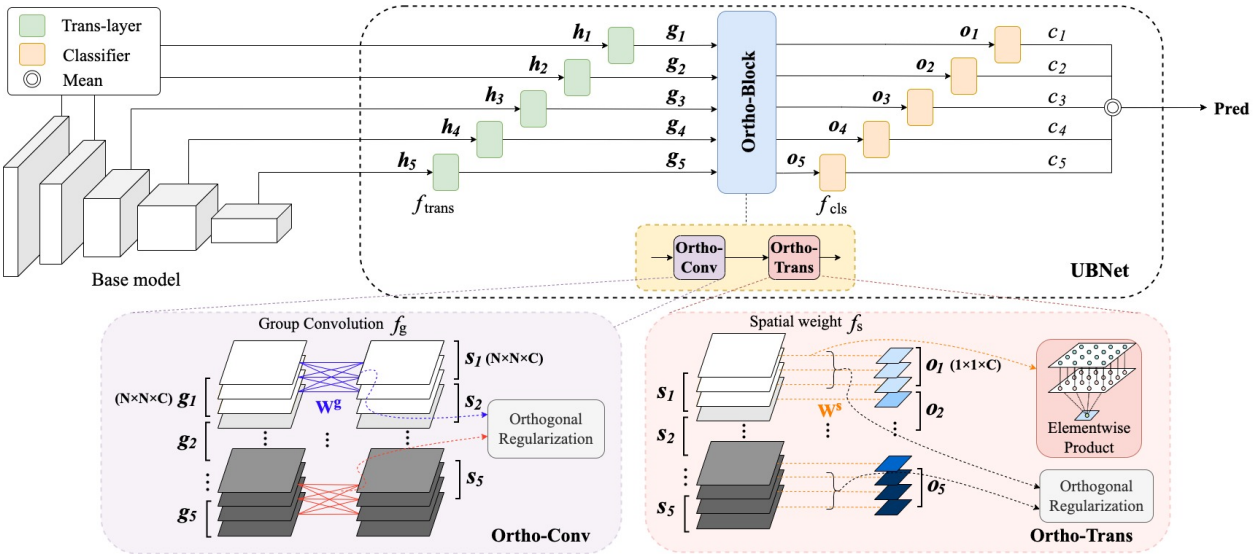


Figure 4. **The architecture of the proposed model, UBNet.** UBNet takes the hierarchical features captured by the base model as input. The Trans-layers set all the $\mathbf{h}_l$ be the same size. Then, all the concatenated $\mathbf{g}_l$ activated through the Ortho-block in which Ortho-Conv and Ortho-Trans layers encode multi-variant features. From the output of the Ortho-block, each classifier outputs confidence scores for each low-to-high feature. They are averaged for the final prediction. In this figure, we use $L = 5$ for simplicity.

learned to be as orthogonal as possible, each hierarchical feature becomes decorrelated. The decorrelation improves the feature expressiveness, which contributes to distributing the feature dependence of the model. To impose the orthogonality, we set the objective function with an additional regularization term:

$$\min_{\theta_f} \mathcal{L}_c\left(\mathbf{y}_{\text{target}}, f(\mathcal{X}_{\text{train}}; \theta_f)\right) + \lambda \mathcal{L}_o\left(\theta_o\right), \tag{5}$$

where f denotes the UBNet (from f_{trans} to prediction) parametrized by θ_f. $\mathcal{L}_o$ stands for the orthogonal regularization loss, defined as

$$\mathcal{L}_o(\theta_o) = \frac{1}{2}\left\|\theta_o^\top \theta_o - I\right\|_F^2, \tag{6}$$

where $\theta_o = [\mathbf{W}_1, \cdots, \mathbf{W}_L]$ is the collection of weights in the Ortho-Block f_o. Each $\mathbf{W}_l$ is the conv-parameters corresponding to the l-th group features $\mathbf{g}_l$. Two orthogonal regularizations are applied here, one for Ortho-Conv and the other for Ortho-Trans, where $\mathbf{W}_l$ indicates $\mathbf{W}_l^g \in \mathbb{R}^{F_g \times F_g \times C}$ for f_g and $\mathbf{W}_l^s \in \mathbb{R}^{F_s \times F_s \times C}$ for f_s, respectively. F_g and F_s denote the size of conv-filters in f_g and f_s, respectively. The (i,j)-component $[\theta_o^\top \theta_o]_{ij}$ of $\theta_o^\top \theta_o \in \mathbb{R}^{L \times L}$ is the inner product of $\mathbf{W}_i$ and $\mathbf{W}_j$, and hence the alternated loss term $\mathcal{L}_o$ measures the cosine similarity between the weights of each group features. By forcing $\theta_o^\top \theta_o$ to be close to the identity matrix, this regularization explicitly forces the network so that the hierarchical features are maintained until the prediction layer, spanning the wider feature space of the image dataset.

Following the Ortho-Block, fully-connected layers (classifier f_{cls} in Fig. 4) are attached to each activated feature group $\mathbf{o}_l$ in order to attain confidence scores $\mathbf{c}_1, \ldots, \mathbf{c}_L \in \mathbb{R}^K$ from each hierarchical features (corresponding to layer l) across K classes. We get our confidence scores $\mathbf{c}$ by taking average over $\mathbf{c}_l$, that is, $\mathbf{c} = \sum_l \mathbf{c}_l / L$, where each group feature equally contributes. All the trainable parameters f_{trans}, f_g, f_s, and f_{cls} are learned end-to-end.

5. Experiments

In this section, we present our extensive experiments on multiple datasets, namely, CelebA-HQ [14], UTKFace [25], and 9-Class ImageNet [11] to empirically verify the effectiveness of the proposed method.

Dataset Protocol. For CelebA-HQ and UTKFace datasets, we follow the 'extreme bias' setting, slightly modified from [2]. Basically, we divide each dataset into two completely biased sets, referred to as 'extreme bias (EB)'. On these two datasets, all images with {(female, long hair), (male, short hair)} belong to EB1, while the opposite images with {(female, short hair), (male, long hair)} are assigned to EB2. Then, we train on one and evaluate on the other to estimate the model's generalizability.

Under the *unknown bias* setting, however, this configuration does not work as intended, since the target and bias variables become exactly the same. In other words, from the model's perspective, classifying the gender and the hair length becomes not distinguishable. Thus, we slightly modify the extreme bias design to mix a small number of the

opposite samples; namely, 'utmost bias (UB)' sets. UB1, for instance, consists of all the EB1 images and $\alpha \times |EB1|$ number of randomly sampled EB2 images where $|.|$ denotes the number of data. This scenario informs the model there are some cases that 'long hair' does not necessarily mean the target class 0 (female), for example. UB is closer to reality than EB since it is rare to have two variables of which distribution is completely identical in a dataset.

In addition to these utmost biased sets, we also evaluate on an unbiased 'test set'. The bias planting protocol [2] that originally proposed to randomly sub-sample for the bias attribute is not directly applicable to our unknown bias setting, so we modify it to uniformly distribute the samples.

For ImageNet, we follow the evaluation protocol of [3]. Specifically, they modified the 9-Class ImageNet [11] (203 classes sub-sampled from ImageNet grouped to 9 superclasses: dog, cat, frog, turtle, bird, primate, fish, crab, insect) to balance the ratios of sub-class images in each superclass. They adopted weighted unbiased accuracy, giving higher weight on rarer pairs of object and texture (*e.g.*, turtle on highway).

Competing models. Unbiased modeling methods under *labeled bias* setting are not comparable, since they require exhaustive labeling on biased variables. In contrast, unbiased models under *known bias* might be applied to *unknown bias*. Therefore, for the CelebA-HQ and UTKFace dataset, we compare the robustness of the proposed method to that of unbiased models, HEX [22], Rebias [3], and LfF [16].

On ImageNet, we compare against StyleisedIN [7], LearnedMixin [5], RUBi [4], Rebias [3], and LfF [16]. Although LearnedMixin and RUBi are designed to unlearn bias for visual question answering task, their object functions were applied in Rebias [3]. We follow the setup in [3] for our comparison.

We use VGG11 [19] as base model for CelebA-HQ and ResNet18 [9] for UTKFace and 9-Class ImageNet. We firstly train the base model then optimize UBNet with the base model's parameters frozen. The layer l is sampled whenever the size of the feature maps is halved ($L = 5$ for both base models). For each experiment, we use Adam optimizer [13] and grid search learning rate (initial value and decay schedule), stopping criterion, and batch size. More implementation details are provided in the supplementary material. For hyperparameters in the competing models, we follow the same setting as presented in the original paper and only apply our data pre-processing for a fair comparison. We repeat each experiment three times and report the average score, unless noted otherwise.

5.1. CelebA-HQ

Dataset and Experimental Settings. CelebA-HQ dataset [14] is composed of 30K high-resolution face images, labeled with gender. In addition, we manually annotated another binary attribute 'hair length', {short, long} for all samples. We excluded 4,518 images (15.06 %) if the hair length is not seen or intermediate. To collect more samples with rarer combinations (male + long hair, female + short hair), we supplement samples from CelebA [15]. The constructed biased dataset consists of 26,851 images in total, where 1,369 of them are from CelebA. Fig. 5 illustrates the two extreme datasets created from CelebA-HQ.

We randomly split the EB1 to train-EB1 and val-EB1. Then, we mix a small portion of images from EB2 (with $\alpha = 0.005$) to create UB1, and use the remainder of EB2 as val-EB2. We train the model on UB1, then evaluate on val-EB1 and val-EB2. More details about the biased CelebA-HQ and concrete train-validation configuration are in the supplementary material.

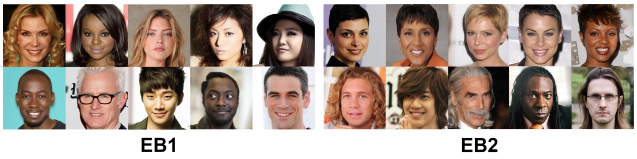


Figure 5. **Extreme bias sets on CelebA-HQ.** EB1 consists of female + long hair and male + short hair, while EB2 consists of male + long hair and female + short hair.

Results and Discussion. In Tab. 1, we report classification accuracy on the val-EB1/2 as well as the test set. The proposed method exhibits more robust results than other models. Mitigating the model's over-reliance on the bias, UBNet shows slightly degraded accuracy on biased val-EB1.

Method	Base Model	HEX	Rebias	LfF	UBNet
Acc (EB1)	**99.38(±0.31)**	92.50(±0.67)	99.05(±0.13)	93.25(±4.61)	99.18(±0.18)
Acc (EB2)	51.22(±1.73)	50.85(±0.37)	55.57(±1.43)	56.70(±6.69)	**58.22(±0.64)**
Acc (test)	75.30(±0.93)	71.68(±0.50)	77.31(±0.71)	74.98(±4.16)	**78.70(±0.24)**

Table 1. **Results on CelebA-HQ.** Classification accuracy on validation set of EB1 and EB2, and on Test set, respectively, for the model trained on UB1 training set.

5.2. UTKFace

Dataset and Experimental Settings. UTKFace [26] consists of 20K face images annotated with age, gender, and skin tone. We evaluate 1) skin tone prediction with gender bias and 2) gender prediction with skin tone bias. Age is not used since the annotation pairs for age is imbalanced. For the details of the overall data distribution of UTKFace, consult with the supplementary material.

For skin tone prediction with gender bias, we split the images into extremely biased sets, illustrated in Fig. 6. Then, we add EB2 image samples (with up to $\alpha = 0.2$) to EB1, and name it UB1. UB2 is created in a similar manner. We train either on UB1 or on UB2, and evalu-

ate on the other. Also, we evaluate both models on an unbiased test set, composed of 300 images for each pair of {gender, skin tone}, thereby 1,200 images in total. The gender prediction with skin tone bias scenario is performed in the same way.

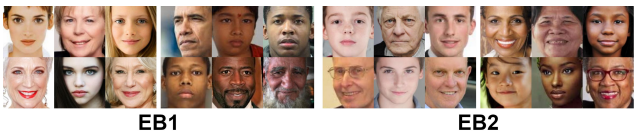


Figure 6. **Extreme bias sets on UTKFace.** EB1 consists of female + bright skin tone and male + dark skin tone and EB2 consists of male + bright skin tone and female + dark skin tone.

Results and Discussion. Table 2 summarizes the experimental results on UTKFace. On the evaluation for the skin tone prediction with the gender bias, UBNet outperforms all other competing models. The proposed method also exhibits the most satisfactory accuracy compared to competing models for gender prediction with skin tone bias.

	Trained on UB1		Trained on UB2	
	UB2	Test	UB1	Test
Skin tone Prediction with Gender Bias				
Base	77.46(±0.55)	82.97(±0.39)	80.58(±0.37)	85.44(±0.32)
HEX	79.35(±0.17)	85.07(±0.46)	80.82(±0.15)	85.88(±0.27)
Rebias	78.70(±0.62)	83.39(±1.22)	80.06(±1.46)	85.41(±1.23)
LfF	77.08(±0.71)	81.67(±0.14)	78.16(±0.97)	84.00(±1.00)
UBNet	**83.67(±1.05)**	**87.25(±0.82)**	**84.29(±1.24)**	**87.94(±0.80)**
Gender Prediction with Skin tone Bias				
Base	80.97(±0.79)	86.67(±0.38)	81.43(±0.11)	85.94(±0.77)
HEX	80.69(±0.80)	86.51(±0.56)	81.01(±1.57)	86.75(±1.57)
Rebias	82.02(±0.84)	86.24(±0.31)	82.02(±0.90)	86.39(±0.51)
LfF	78.82(±0.31)	83.67(±0.73)	82.14(±1.37)	85.36(±0.99)
UBNet	**82.07(±1.55)**	**87.75(±0.58)**	**82.69(±0.80)**	**87.36(±0.46)**

Table 2. **Results on UTKFace.** Skin tone prediction with gender bias and gender prediction with skin tone bias results.

5.3. ImageNet

One might argue that the experiments on CelebA-HQ and UTKFace are designed unfavorably to HEX and Rebias, since they are specially designed for texture bias. For this reason, we conduct experiments on texture bias, following the same setting by Rebias [3].

Dataset. We evaluate the generalizability of our model on the balanced 9-Class ImageNet, presented by Rebias [3]. Figure 7 illustrates a few examples, frequent cases (*e.g.*, a frog in a swamp) in the top row and less common images (*e.g.*, a frog on a hand) in the bottom row. Rebias [3] designed this dataset, arguing this correlation between background and object makes texture bias in the train set.

Results and Discussion. Table 3 summarizes the biased and unbiased classification accuracy for the same validation set. The unbiased accuracy [3] puts higher weights on

Figure 7. **Textually biased and unbiased ImageNet.** The examples are sampled from 9-Class ImageNet [11].

samples with unusual texture-class combinations. The biased accuracy is the regular accuracy, the number of correct samples divided by the total number of samples.

Surprisingly, we observe that the proposed method achieves the state-of-the-art. This is unexpectedly notable, since it even outperforms Rebias, which was specially designed to tackle texture bias, under the experimental setting and on the dataset they designed. This verifies that UBNet is capable of discovering *unknown bias* effectively enough to compete with a model designed for *known bias* settings.

Metric	Base model	SI	LM	RUBi	Rebias	LfF	UBNet
Biased	90.8	88.4	64.1	90.5	**91.9**	89.0	**91.9**
Unbiased	88.8	86.6	62.7	88.6	90.5	88.2	**91.5**

Table 3. **Results on 9-class ImageNet.** The results except for UBNet are referenced from [3]. SI denotes StyleisedIN and LM denotes LearnedMixin.

5.4. Ablation Study

We conduct 4 ablation studies to display the significance of all the sub-components exploited in UBNet. Experimental settings are the same as Sec. 5.1 unless noted otherwise.

Ablation Study on Sub-components. We conduct ablation studies to see the effects of hierarchical features, group convolution, and orthogonal regularization presented in Sec. 4. Table 4 exhibits the performance on an unbiased test set, incrementally adding each component. The hierarchical features contribute the most, although other components also improve the performance notably.

Hierarchical feature		✓	✓	✓
Group convolution			✓	✓
Othogonal regularization				✓
Acc (EB1)	**99.35**	99.30	99.08	99.18
Acc (EB2)	52.87	56.13	56.78	**58.22**
Acc (test)	76.11	77.72	77.93	**78.70**

Table 4. **Ablation Study on Sub-components.** The sub-components presented in Sec. 4 are applied step by step. The model with none of them applied is the last phase in Fig. 4 from $\mathbf{h}_L$ to $\mathbf{c}_L$ in the absence of orthogonal regularization.

Ablation Study on f_g. We conduct an ablation study with and without f_g. Table 5 exhibits Ortho-Conv f_g contributes to performance improvement. Some lower-level

features might function just as intermediate representations needed for making high-level features and hence be less useful to understand the image. f_g gives more attention to meaningful features for inferring the target label.

Method	Acc (EB1)	Acc (EB2)	Acc (test)
UBNet (w/o f_g)	**99.32(±0.08)**	51.80(±1.39)	75.56(±0.73)
UBNet (with f_g)	99.18(±0.18)	**58.22**(±0.64)	**78.70**(±0.24)

Table 5. **Ablation study on f_g.** UBNet (with f_g) is UBNet in Tab. 1 and UBNet (w/o f_g) denotes the model is Ortho-Conv.

Ablation Study on Hierarchical Features. We compare the results by excluding the lowest-level features one by one from the five hierarchical features extracted from VGG11. According to Fig. 8, it is obvious that the multi-level features contribute to the performance on the unbiased test set. The number of parameters slightly increases as more hierarchical features are used, but it is difficult to see that the parameters are the main factor for performance improvement because the difference is subtle.

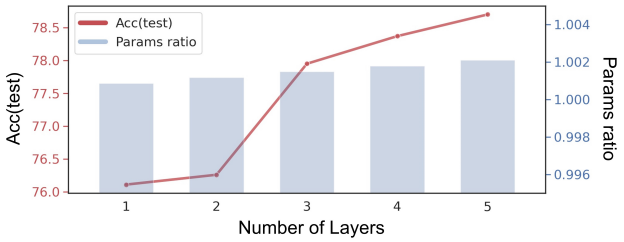


Figure 8. **Ablation Study on Hierarchical Features.** The line plot denotes the accuracy on the uniformly distributed test set, and the bar plot denotes the ratio of the number of parameters relative to the base model.

Degree of Bias (DoB) $\Delta r^{[l]}$ after f_g and f_s. The lower-level features could be biased after additional layers, although originally unbiased. We evaluate every difference between DoB of h_n and that of o_n ($n = 1, \ldots, 5$) of Fig. 3. The subtle difference of each pair in Tab. 6 indicates that the lower-level features still remain the DoB as expected.

Layer depth	1	2	3	4	5
ΔDoB	0.024	0.012	0.027	0.021	-0.0005

Table 6. **Degree of Bias.** ΔDoB = DOB(h_n) - DOB(o_n).

6. Related Work

Labeled bias. Alvi *et al*. [2] jointly trained a multi-headed model, and minimized the confusion losses of the tasks other than the primary classification for the shared feature representation to be invariant. Kim *et al*. [12] employed an additional network to predict the bias distribution and trained the main network adversarially against bias-oriented network. They formulated regularization loss based on mutual information. Motivated by lower face recognition errors for certain cohorts in some demographic groups, Gong *et al*. [8] proposed to learn a disentangled representation to unbias face recognition and demographic attribute estimation. Adeli *et al*. [1] defined a surrogate loss to predict the bias while quantifying the statistical dependence with respect to target bias based on Pearson correlation. Dhar *et al*. [6] proposed a feature-based adversarial unbiasing, with a discriminator training strategy that discourages a network from encoding protected attribute information.

Known bias. Geirhos *et al*. [7] showed that CNNs trained on ImageNet are strongly biased towards recognizing textures rather than shape variations. They constructed Stylized-ImageNet dataset which makes the model be able to learn shape-based representations. Rebias [3] encouraged de-biased representation to be different from a set of textually biased representations through applying Hilbert-Schmidt independence criterion. The intentionally biased model towards texture is reproduced by reducing the receptive fields. HEX [22] unlearned texture bias or subtle color changes by utilizing the neural gray-level co-occurrence matrix. The biased features were encouraged to be removed through the projection in the learned representations.

Nam *et al*. [16] posed easily learned features as malignant biases, addressing them by adjusting sample weights at training. Although this definition of bias falls into neither *labeled* nor *known*, it is different from our *unknown bias*.

7. Conclusion

Excessive dependence on the distribution of the training data causes a machine learning model to be unstable on the unseen data. We specifically mitigate the case of model biasing, when data is distributed severely biased towards certain attributes. Contrary to the previous studies that address the heuristically defined bias, we deal with the *unknown bias* setting. As unknown biases cannot be quantitatively measured, we tackle this issue by exploiting multi-variant and unbiased representations via *hierarchical features* and *orthogonal regularization*. The investigations on the CNN's representations support our motivation. Extensive evaluations demonstrate our UBNet contributes to the model's robustness on generalization, and further ablation studies show the potency of the proposed method on unbiased learning.

Acknowledgement. Myeongho Jeon and Daekyoung Kim contributed equally. Corresponding authors are Joonseok Lee, Myoungjoo Kang. Joonseok Lee was supported by SNU new faculty startup fund and research grants NRF [2021H1D3A2A03038607, 2022R1C1C1010627] and IITP [2021-0-01778]. Myungjoo Kang was supported by NRF [2021R1A2C3010887] and MSIT/IITP [1711117093, 2021-0-00077].

References

[1] Ehsan Adeli, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Li Fei-Fei, Juan Carlos Niebles, and Kilian M Pohl. Representation learning with statistical independence to mitigate bias. In *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. 1, 8

[2] Mohsan Alvi, Andrew Zisserman, and Christoffer Nellåker. Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In *Proc. of the European Conference on Computer Vision (ECCV) Workshops*, 2018. 1, 5, 6, 8

[3] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *Proc. of the International Conference on Machine Learning (ICML)*, 2020. 1, 2, 6, 7, 8

[4] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 6

[5] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv:1909.03683*, 2019. 6

[6] Prithviraj Dhar, Joshua Gleason, Aniket Roy, Carlos D Castillo, and Rama Chellappa. PASS: Protected attribute suppression system for mitigating bias in face recognition. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 8

[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv:1811.12231*, 2018. 1, 2, 3, 6, 8

[8] Sixue Gong, Xiaoming Liu, and Anil K Jain. Jointly de-biasing face recognition and demographic attribute estimation. In *Proc. of the European conference on computer vision (ECCV)*, 2020. 1, 8

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 6

[10] Gao Huang, Shichen Liu, Laurens Van der Maaten, and Kilian Q Weinberger. CondenseNet: An efficient densenet using learned group convolutions. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[11] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv:1905.02175*, 2019. 5, 6, 7

[12] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 8

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 6

[14] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5, 6

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015. 6

[16] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 6, 8

[17] Inioluwa Deborah Raji, Timnit Gebru, Margaret Mitchell, Joy Buolamwini, Joonseok Lee, and Emily Denton. Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proc. of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020. 1

[18] Ignacio Serna, Alejandro Peña, Aythami Morales, and Julian Fierrez. InsideBias: Measuring bias in deep networks and application to face gender biometrics. In *Proc. of the International Conference on Pattern Recognition (ICPR)*, 2021. 2, 3

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 3, 6

[20] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *Domain adaptation in computer vision applications*, pages 37–55. Springer, 2017. 1

[21] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 1

[22] Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. *arXiv:1903.06256*, 2019. 1, 6, 8

[23] Di Xie, Jiang Xiong, and Shiliang Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4

[24] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. of the European conference on computer vision (ECCV)*, 2014. 4

[25] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[26] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6