Examining CNN Representations With Respect To Dataset Bias

Quanshi Zhang[†], **Wenguan Wang**[†,‡], **and Song-Chun Zhu**[†]
[†]University of California, Los Angeles [‡]Beijing Institute of Technology

Abstract

Given a pre-trained CNN without any testing samples, this paper proposes a simple yet effective method to diagnose feature representations of the CNN. We aim to discover representation flaws caused by potential dataset bias. More specifically, when the CNN is trained to estimate image attributes, we mine latent relationships between representations of different attributes inside the CNN. Then, we compare the mined attribute relationships with ground-truth attribute relationships to discover the CNN's blind spots and failure modes due to dataset bias. In fact, representation flaws caused by dataset bias cannot be examined by conventional evaluation strategies based on testing images, because testing images may also have a similar bias. Experiments have demonstrated the effectiveness of our method.

Introduction

Given a convolutional neural network (CNN) that is pre-trained to estimate image attributes (or labels), how to diagnose black-box knowledge representations inside the CNN and discover potential representation flaws is a crucial issue for deep learning. In fact, there is no theoretical solution to identifying good and problematic representations in the CNN. Instead, people usually just evaluate a CNN based on the accuracy obtained using testing samples.

In this study, we focus on representation flaws caused by potential bias in the collection of training samples (Torralba and Efros 2011). As shown in Fig. 1, if an attribute usually co-appears with certain visual features in training samples, then the CNN may be learned to use the co-appearing features to represent this attribute. When the used co-appearing features are not semantically related to the target attribute, we consider these features as biased representations. This idea is related to the disentanglement of the local, bottom-up, and top-down information components for prediction (Wu, Xia, and Zhu 2007; Yang, Wu, and Zhu 2009; Wu and Zhu 2011). We need to clarify correct and problematic contexts for prediction. CNN representations may be biased even when the CNN achieves a high accuracy on testing samples, because testing samples may have a similar bias.

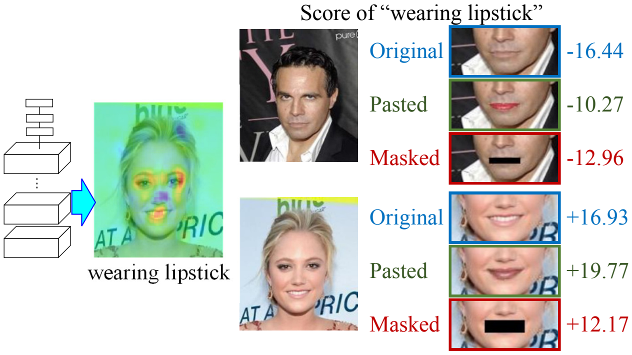


Figure 1: Biased representations in a CNN. Considering potential dataset bias, a high accuracy on testing images cannot always ensure that a CNN learns correct representations. The CNN may use unreliable co-appearing contexts to make predictions. For example, we manually modify mouth appearances of two faces by masking mouth regions or pasting another mouth, but such modifications do not significantly change prediction scores for the *lipstick* attribute. We show heat maps of inference patterns of the *lipstick* attribute, where patterns with red/blue colors are positive/negitive with the attribute score. The CNN mistakenly considers unrelated patterns as contexts to infer the lipstick. We propose a method to automatically discover such biased representations from a CNN without any testing images.

In this paper, we propose a simple yet effective method that automatically diagnoses representations of a pre-trained CNN without given any testing samples. *I.e.,* we only use training samples to determine the attributes whose representations are not well learned. We discover blind spots and failure modes of the representations, which can guide the collection of new training samples.

Intuition, self-compatibility of network representations: Given a pre-trained CNN and an image I, we use the CNN to estimate attribute A for I. We also mine inference patterns[1] of the estimation result, which are hidden in conv-

[1] We regard a neural pattern as a group of units in a channel of a conv-layer's feature map, which are activated and play a crucial role in the estimation of the attribute A.

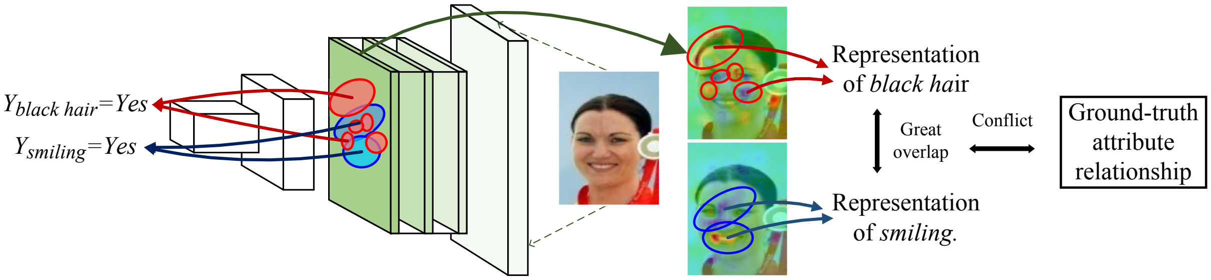

$Y_{black\ hair}$=Yes
$Y_{smiling}$=Yes
Representation of *black hair*
Great overlap
Conflict
Ground-truth attribute relationship
Representation of *smiling*.

Figure 2: Overview of the method. Given a biased dataset for training where the *smiling* and *black hair* attributes usually appear on faces with certain appearances of eyes or noses, the CNN may mistakenly use eye or nose features to represent the two attributes. Biased representations are difficult to discover when testing samples are also biased. In this study, we mine relationships between attributes. Conflicts between the mined and ground-truth relationships indicate potential representation problems.

layers of the CNN. We can regard the mined inference patterns as exact representations of the attribute A in the CNN. Then, based on inference patterns, we compute the relationship between each pair of attributes (A_i, A_j), *i.e.* identifying whether A_i is positively/negatively/not related to A_j.

The intuition is simple, *i.e.* according to human's common sense, we set up several ground-truth relationships between some pairs of attributes as rules to diagnose CNN representations. The mined attribute relationships should well fit the ground truth; otherwise, the representation is probably not well learned. Let us take a CNN that is learned to estimate face attributes for example. As shown in Fig. 2, the *smiling* attribute is supposed to be represented by features (patterns), which appear on the mouth region in conv-layers. Whereas, the *black hair* attribute should be inferred by features extracted from hairs. Therefore, the attribute relationship "*smiling* is not related to the *black hair*" is trustworthy enough to become a ground truth. However, the CNN may use eye/nose features to represent the two attribute, because these attributes always co-appear with specific eye/nose appearances in a biased dataset. Thus, we will mine a specific relationship between the two attributes, which conflicts with the ground truth.

Our method: Given a pre-trained CNN, we mine relationships between each pair of attributes according to their inference patterns. Then, we annotate some ground-truth attribute relationships. For example, the *heavy makeup* attribute is positively related to the *attractive* attribute; *black hair* and *smiling* are not related to each other. We compute the Kullback-Leibler (KL) divergence between the mined relationships and ground-truth relationships to discover attributes that are not well learned, including both blind spots and failure modes of attribute representations.

In fact, how to define ground-truth relationships is still an open problem. We can ask different people to label attribute relationships in their personal opinions to approach the ground truth. More importantly, our method is compatible with various types of ground-truth distributions. People can define their ground truth *w.r.t.* their tasks as constraints to examine the network. Thus, our method is a flexible and convincing way to discover representation bias at the level of human cognition.

The annotation cost of our method is usually much lower than end-to-end learning of CNNs. Our annotation cost is $O(n^2)$, where n denotes the number of attribute outputs. In contrast, it usually requires thousands or millions of samples to learn a new CNN in real applications.

Why is the proposed method important? As a complement to using testing samples for evaluation, our zero-shot diagnosis of a CNN is of significant values in applications:

- A high accuracy on potentially biased testing samples cannot prove correct representations of a CNN.
- Potential bias cannot be fully avoided in most datasets. Especially, some attributes (*e.g. smiling*) mainly describe specific parts of images, but the dataset (Liu et al. 2015; Patterson et al. 2014) only provides image-level annotations of attributes for supervision without specifying regions of interests, which makes the CNN more sensitive to dataset bias.

  More crucially, the level of representation bias is not necessary to be proportional to the dataset bias level. We need to diagnose the actual CNN representations.
- In conventional studies, correcting representation flaws caused by either dataset bias or the over-fitting problem is a typical long-tail problem. If we blindly collect new training samples without being aware of failure modes of the representation, it would require massive new samples to overcome the bias problem. Our method provides a new perspective to solve the long-tail problem.
- Unlike methods of CNN visualization/analysis (Zeiler and Fergus 2014; Mahendran and Vedaldi 2015; Simonyan, Vedaldi, and Zisserman 2014; Ribeiro, Singh, and Guestrin 2016) that require people to one-by-one check the representation of each image, our method discovers all biased representations in a batch.

Contribution: In this study, to the best of our knowledge, we, for the first time, propose a method to discover potentially biased representations hidden in a pre-trained CNN without testing samples. Our method mines blind spots and failure modes of a CNN in a batch manner, which can guide the collection of new samples. Experiments have proved the effectiveness of the proposed method.

Related work

Visualization of CNNs: In order to open the black box of a CNN, many methods (Zeiler and Fergus 2014; Mahendran and Vedaldi 2015; Simonyan, Vedaldi, and Zisserman 2014; Aubry and Russell 2015; Liu, Shen, and van den Hengel 2015; Dosovitskiy and Brox 2016) have been developed to visualize and analyze patterns of response units in a CNN. Some methods (Zeiler and Fergus 2014; Mahendran and Vedaldi 2015; Simonyan, Vedaldi, and Zisserman 2014) back-propagate gradients *w.r.t.* a given unit to pixel values of an image, in order to obtain an image that maximizes the score of the unit. These techniques mainly visualize simple patterns. As mentioned in (Farhadi et al. 2009), attributes are an important perspective to model images, but it is difficult to visualize a complex attribute (*e.g.* the *attractive* attribute).

Given a feature map produced by a CNN, Dosovitskiy *et al.* (Dosovitskiy and Brox 2016) trained a new up-convolutional network to invert the feature map to the original image. Similarly, this approach was not designed for the visualization of a single attribute output.

Interpreting semantic meanings of CNNs: Going beyond the "passive" visualization of neural patterns, some studies "actively" retrieve mid-level patterns from conv-layers, which potentially corresponds to a certain object/image part. Zhou *et al.* (Zhou et al. 2015; 2016) mined patterns for "scene" semantics from feature maps of a CNN. Simon *et al.* discovered objects (Simon and Rodner 2015) from CNN feature maps in an unsupervised manner, and retrieved patterns for object parts in a supervised fashion (Simon, Rodner, and Denzler 2014). Zhang *et al.* (Zhang et al. 2016) used a graphical model to organize implicit mid-level patterns mined from a CNN, in order to explain the pattern hierarchy inside conv-layers in a weakly-supervised manner. (Goyal et al. 2016) used a gradient-based method to interpret visual question-answering models. Zhang *et al.* (Zhang et al. 2018) transformed CNN representations to an explanatory graph, which represents the semantic hierarchy hidden inside a pre-trained CNN.

Model diagnosis: Many methods have been developed to diagnose representations of a black-box model. (Adler et al. 2016) extracted key features for model outputs. The LIME method proposed by Ribeiro *et al.* (Ribeiro, Singh, and Guestrin 2016) and gradient-based visualization methods (Fong and Vedaldi 2017; Selvaraju et al. 2017) extracted image regions that were responsible for each network output, in order to interpret the network representation.

Unlike above studies diagnosing representations for each image one by one, many approaches aim to evaluate all potential attribute/label representations for all images in a batch. Lakkaraju *et al.* (Lakkaraju et al. 2017) and Zhang *et al.* (Zhang et al. 2017b; 2017a) explored unknown knowledge hidden in CNNs via active annotations and active question-answering. Methods of (Bansal, Farhadi, and Parikh 2014; Zhang et al. 2014) computed the distributions of a CNN's prediction errors among testing samples, in order to summarize failure modes of the CNN. However, we believe that compared to (Bansal, Farhadi, and Parikh 2014; Zhang et al. 2014), it is of larger value to explore evidence of failure cases from mid-layer representations of a CNN. (Ross, Hughes, and Doshi-Velez 2017) required people to label dimensions of input features that were related to each output according to common sense, in order to learn a better model. Hu *et al.* (Hu et al. 2016) designed some logic rules for network outputs, and used these rules to regularize the learning of neural networks. In our research, we are inspired by Deng *et al.* (Deng et al. 2014), which used label graph for object classification. We use ground-truth attribute relationships as logic rules to harness mid-layer representations of attributes. (Wu, Xia, and Zhu 2007; Yang, Wu, and Zhu 2009; Wu and Zhu 2011) tried to isolate and diagnose information from local, bottom-up, or top-down inference processes. More specially, (Wu and Zhu 2011) proposed to separate implicit local representations and explicit contextual information used for prediction. Following this direction, this is the first study to diagnose unreliable contextual information from CNN representations *w.r.t.* dataset bias.

Active learning: Active learning is a well-known strategy for detecting "unknown unknowns" of a pre-trained model. Given a large number of unlabeled samples, existing methods mainly select samples on the decision boundary (Vijayanarasimhan and Grauman 2011) or samples that cannot well fit the model (Long and Hua 2015; Zhang et al. 2017b), and require human users to label these samples.

Compared to active-learning approaches, our method does not require any additional unlabeled samples to test the model. More crucially, our method looks deep inside the representation of each attribute to mine attribute relationships; whereas active learning is closer to black-box testing of model performance. As discussed in (Suh, Zhu, and Amershi 2016), unless the initial training set contains at least one sample in each possible mode of sample features, active learning may not exhibit high efficiency in model refinement.

Algorithm

Problem description

We are given a CNN that is trained using a set of images $\mathbf{I}$ with attribute annotations. The CNN is designed to estimate n attributes of an image, denoted by $A_1, A_2, \ldots, A_n$. Meanwhile, we also have a certain number of ground-truth relationships between different attributes, denoted by a relationship graph $G^* = (\{A_i\}, \mathbf{E}^*)$. Each edge $(A_i, A_j) \in \mathbf{E}^*$ represents the relationship between A_i and A_j. Note that it is **not** necessary for G^* to be a complete graph. We only select trustworthy relationships as ground truth. The goal is to identify attributes that are not well learned and to discover blind spots and failure modes in attribute representation.

Given an image $I \in \mathbf{I}$, let Y_i^I and $Y_i^{I,*}$ denote the attribute value of A_i estimated by the CNN and the ground-truth annotation for A_i. In order to simplify the notation, we omit superscript I and use notations of Y_i and Y_i^* in most sections, except in Section .

In different applications, people use multiple ways to define attributes (or labels), including binary attributes ($Y_i \in \{-1, +1\}$) and continuous attributes (*e.g.* $Y_i \in [-1, +1]$ and $Y_i \in (-\infty, +\infty)$). We can normalize all these attributes to the range of $Y_i \in (-\infty, +\infty)$ for simplification[2]. To simplify the introduction, without loss of generality, we consider $Y_i^* > 0$ as the existence of a certain attribute A_i; otherwise not. Consequently, we flip the signs of some ground-truth annotations to ensure that we use positive values, rather than negative values, to represent the activation of A_i.

Mining attribute relationships

Attribute representation: Given an image $I \in \mathbf{I}$ and a target attribute A_i, we select the feature map $\mathbf{x}^I$ of a certain conv-layer of the CNN to represent A_i and compute Y_i^I. Since the CNN conducts a series of convolution and ReLu operations on $\mathbf{x}^I$ to compute Y_i^I, we can approximate Y_i^I as a linear combination of neural activations in $\mathbf{x}^I$.

$$Y_i^I \approx (\mathbf{v}_i^I)^T \mathbf{x}^I + \beta_i^I, \qquad \mathbf{v}_i^I = \boldsymbol{\rho}_i \circ \boldsymbol{\nu}_i^I \tag{1}$$

where $\mathbf{v}_i^I$ denotes a weight vector, and β_i^I is a scalar for bias.

In the above equation, parameters $\boldsymbol{\nu}_i^I$ and β_i^I reflect inherent piecewise linear representations of Y_i^I inside the CNN, whose values have been fixed when the CNN and the image are given. We will introduce the estimation of $\boldsymbol{\nu}_i^I$ and β_i^I later. The target parameter here is $\boldsymbol{\rho}_i \in \{0, 1\}^N$, which is a sparse mask vector. It means that we select a relatively small number of reliable neural activations from $\mathbf{x}^I$ as inference patterns of A_i and filters out noises. $\circ$ denotes element-wise multiplication between vectors. We can regard $\boldsymbol{\rho}_i$ as a prior spatial distribution of neural activations that are related to attribute A_i. For example, if A_i represents an attribute for noses, then we expect $\boldsymbol{\rho}_i$ to mainly represent nose regions. Note that except $\boldsymbol{\rho}_i$, parameters $\boldsymbol{\nu}_i^I$ and β_i^I are only oriented to image I due to ReLu operations in the CNN.

We can compute the inherent piecewise linear gradient *w.r.t.* $\mathbf{x}^I$, *i.e.* $\boldsymbol{\nu}_i^I$ via gradient back propagation.

$$\boldsymbol{\nu}_i^I = \frac{\partial Y_i}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}^I} = \frac{\partial Y_i}{\partial \mathbf{x}_M} \frac{\partial \mathbf{x}_M}{\partial \mathbf{x}_{M-1}} \cdots \frac{\partial \mathbf{x}_{m+2}}{\partial \mathbf{x}_{m+1}} \frac{\partial \mathbf{x}_{m+1}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}^I} \tag{2}$$

where the CNN contains M conv-layres (including fully-connected layers), and $\mathbf{x}_k$ denotes the output of the k-th conv-layer ($\mathbf{x} \overset{\text{def}}{=} \mathbf{x}_m$ corresponds to the m-th conv-layer). We can further compute the value of β_i^I based on the full representation without pattern selection $Y_i^I = (\boldsymbol{\nu}_i^I)^T \mathbf{x}^I + \beta_i^I$.

Inspired by the LIME method (Ribeiro, Singh, and Guestrin 2016), the loss of mining inference patterns is similar to a Lasso selection:

$$\hat{\boldsymbol{\rho}}_i = \underset{\boldsymbol{\rho}_i}{\text{argmin}}\ \mathbf{E}_{I \in \mathbf{I}}\big[\mathcal{L}(Y_i^I, \boldsymbol{\rho}_i)\big] + \mathbf{L}(\boldsymbol{\rho}_i) \tag{3}$$

where $\mathcal{L}(Y_i^I, \boldsymbol{\rho}_i)$ measures the fidelity of the representation on image I, and $\mathbf{L}(\boldsymbol{\rho}_i)$ denotes the representation complexity. We can simply formulate $\mathcal{L}(Y_i^I, \boldsymbol{\rho}_i) = [(\mathbf{v}_i^I)^T \mathbf{x}^I + \beta_i^I - Y_i^I]^2$, and $\mathbf{L}(\boldsymbol{\rho}_i) = \lambda \|\boldsymbol{\rho}_i\|_1$, where $\|\cdot\|_1$ denotes L-1 norm, and λ is a constant. Based on the above equation, $\hat{\boldsymbol{\rho}}_i$ can be directly estimated using a greedy strategy.

Attribute relationships: For each pair of attributes A_i and A_j, we define a cosine distance $\varpi_{ij}^I \overset{\text{def}}{=} \frac{(\mathbf{v}_j^I)^T \mathbf{v}_i^I}{\|\mathbf{v}_j^I\| \|\mathbf{v}_i^I\|}$ to represent their attribute relationship. If A_i and A_j are positively related, $\mathbf{v}_i$ will approximate to $\mathbf{v}_j$, *i.e.* ϖ_{ij}^I will be close to 1. Similarly, if A_i and A_j are negatively related, then ϖ_{ij}^I will be close to -1. If A_i and A_j are not closely related, then $\mathbf{v}_j$ and $\mathbf{v}_i$ will be almost orthogonal, thus $\varpi_{ji}^I \approx 0$.

The actual representation of an attribute in a CNN is highly non-linear, and the linear representation in Eq. (1) is just a local mode oriented to a specific image I. When we compute the gradient $\boldsymbol{\nu}_i^I = \frac{\partial Y_i}{\partial \mathbf{x}^I}$, the ReLu operation blocks irrelevant information in gradient back-propagation, thereby obtaining a local linear representation. It is possible to cluster $\boldsymbol{\nu}_i^I$ of different images into several local modes of the representation. Expect extreme cases mentioned in (Koh and Liang 2017), these local modes are robust to most small perturbations in the image I.

Diagnosis of CNN representations

Given each image $I \in \mathbf{I}$, we compute ϖ_{ij}^I to represent the relationship between A_i and A_j *w.r.t.* the image I. In this way, we use the distribution of ϖ_{ij}^I among all training images in $\mathbf{I}$, denoted by $\mathbf{Q}(\varpi_{ij}|A_i, A_j)$, to represent the overall attribute relationship[3]. Fig. 3 shows the mined distributions of $\mathbf{Q}(\varpi_{ij}|A_i, A_j)$ for different pairs of attributes.

Besides the observation distribution $\mathbf{Q}$, we also manually annotate a number of ground-truth attribute relationships G^*, and define a distribution for each ground-truth attribute relationship $\mathbf{P}(\varpi_{ij}|A_i, A_j)$. People can label several types of ground-truth relationships for $(A_i, A_j) \in \mathbf{E}^*$, $l_{ij} \in \mathbf{L} = \{L_1, L_2, \ldots\}$, to supervise the diagnosis of CNN representations. Let $(A_i, A_j) \in \mathbf{E}^*$ be labeled with $l_{ij} = L^*$. We assume the ground-truth distribution $\mathbf{P}(\varpi_{ij}|A_i, A_j) \sim \mathcal{N}(\mu_{L^*}, \sigma_{L^*}^2)$ follows a Gaussian distribution. We assume most pairs of attributes are well learned, so we can compute μ_{L^*} and $\sigma_{L^*}^2$ as the mean and the variation of ϖ_{ij}, respectively, among all pairs of attributes that are labeled with L^*. In this way, biased representations correspond to outliers of ϖ_{ij} *w.r.t* the ground-truth distribution.

We can compute the KL-divergence between $\mathbf{P}$ and $\mathbf{Q}$,

[2] Given annotations of continuous attributes $Y_i^* \in (-\infty, +\infty)$, we can define L-2 norm loss $L(Y_i, Y_i^*) = (Y_i^* - Y_i)^2$ to train the CNN. Given annotations of binary attributes for training $Y_i^* \in \{-1, +1\}$, we can use the logistic log loss $L(Y_i, Y_i^*) = \log(1 + \exp(-Y_i \cdot Y_i^*))$ to train the CNN. In this way, Y_i can be considered as an attribute estimation whose range is $(-\infty, +\infty)$.

[3] Without loss of generality, we modify attribute annotations to ensure $Y_i^* = +1$ rather than $Y_i^* = -1$ to indicate the existence of a certain attribute. We find that the CNN mainly extracts common patterns from positive samples as inference patterns to represent each attribute. Thus, we compute distributions $\mathbf{P}$ and $\mathbf{Q}$ for (A_i, A_j) among the samples in which either $Y_i^* = +1$ or $Y_j^* = +1$. Similarly, in Experiment 3, we also ignored samples with $Y_i^* = Y_j^* = -1$ to compute the entropy for the competing method.

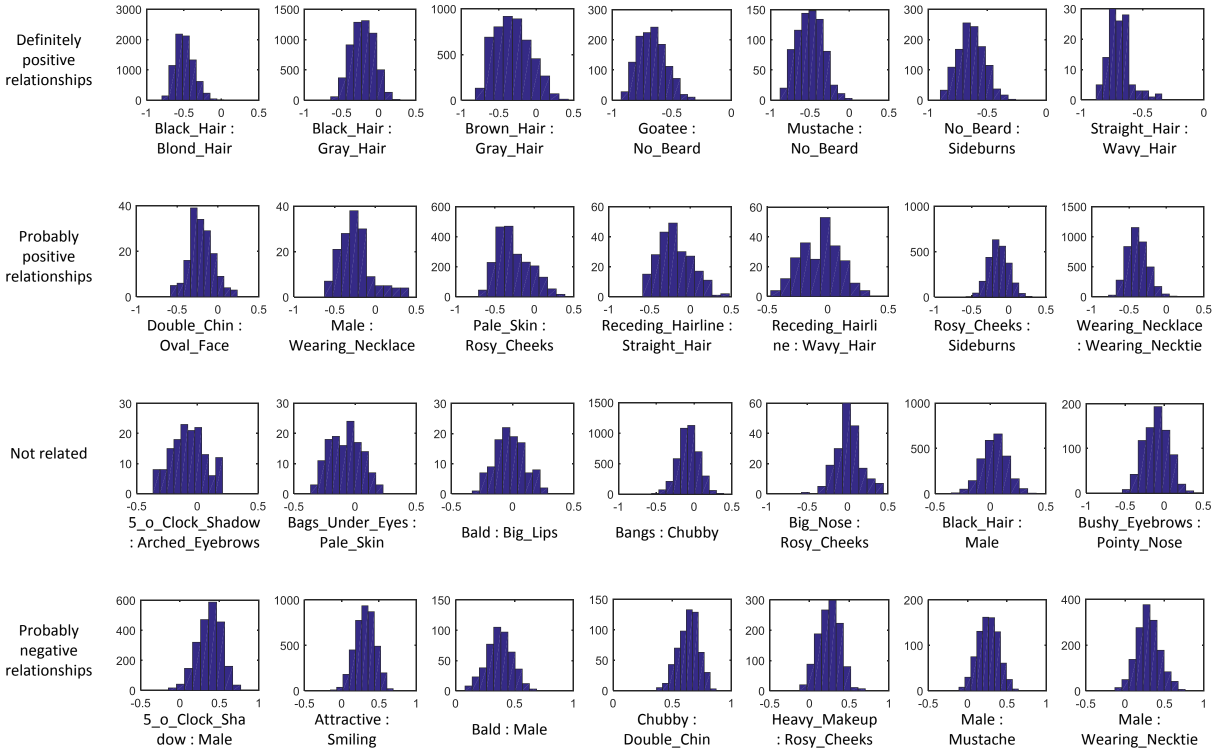


Figure 3: Histograms of ϖ_{ij} to describe distributions of $\mathbf{Q}(\varpi_{ij}|A_i,A_j)$ for different pairs of attributes. The horizontal axis indicates the value of ϖ_{ij}. Note that the vertical axis indicates the number of samples in the histogram, which is not the density of $\mathbf{Q}(\varpi_{ij}|A_i,A_j)$. In the first, second, third, and fourth rows, we show the mined distributions for attribute pairs that are labeled with "definitely positive relationships," "probably positive relationships," "not-related relationships," and "probably negative relationships," respectively.

$\mathbf{KL}(\mathbf{P}\|\mathbf{Q})$, to discover biased representations.

$$\mathbf{KL}_{A_iA_j}=\int_{\Omega}\mathbf{P}(\varpi_{ij}|A_j,A_i)\log\frac{\mathbf{P}(\varpi_{ij}|A_j,A_i)}{\mathbf{Q}(\varpi_{ij}|A_j,A_i)}\mathbf{d}\varpi_{ij} \qquad (4)$$

$$\begin{aligned}\mathbf{KL}_{A_i}&=\sum_{j:(A_i,A_j)\in\mathbf{E}^*}\int_{\Omega}\mathbf{P}(\varpi_{ij},A_j|A_i)\log\frac{\mathbf{P}(\varpi_{ij},A_j|A_i)}{\mathbf{Q}(\varpi_{ij},A_j|A_i)}\mathbf{d}\varpi_{ij} \qquad (5)\\&=\sum_{j:(A_i,A_j)\in\mathbf{E}^*}P(A_j|A_i)\mathbf{KL}_{A_iA_j}\end{aligned}$$

where $P(A_j|A_i)=1/\deg(A_i)$ is a constant given the degree of A_i. We approximately set $\Omega=[-1,1]$, because $\mathbf{P}(\varpi_{ij}|A_j,A_i)\approx 0$ when $|\varpi_{ij}|>1$ in real applications. We believe that if $\mathbf{KL}_{A_i}$ is high, A_i is probably not well learned.

Blind spots & failure modes: Each pair of attributes $(A_i,A_j)\in\mathbf{E}^*$ with a high $\mathbf{KL}_{A_iA_j}$ may have two alternative explanations. The first explanation is that (A_i,A_j) represents a blind spot of the CNN. *I.e.* (A_i,A_j) should be positively/negatively related to each other according to the ground-truth, but the CNN has not learned many inference patterns that are shared by both A_i and A_j. In this case, the CNN does not encode the inference relationship between A_i and A_j.

The alternative explanation is that (A_i,A_j) represents a failure mode. If the mined relationship is that A_i is strongly positively related to A_j, which conflicts with the ground-truth relationship. Then, samples with opposite ground-truth annotations for A_i and A_j, $Y_i^*\cdot Y_j^*<0$ may correspond to a failure mode in attribute estimation. Note that these samples belong to two modes, *i.e.* the modes of $Y_i^*>0, Y_j^*<0$ and $Y_i^*<0, Y_j^*>0$. We simply select the mode with fewer samples as a failure mode. Similarly, if the CNN incorrectly encodes a negative relationship between (A_i,A_j), then we select a failure mode from candidates of $(Y_i^*>0,Y_j^*>0)$ and $(Y_i^*<0,Y_j^*<0)$.

In practise, we determine blind spots and failure modes as follows. Given a pair of attributes (A_i,A_j) with a high $\mathbf{KL}_{A_iA_j}$, if $|\mathbf{E}_I[\varpi_{ij}^I]|<0.2$ and $|\mathbf{E}_I[\varpi_{ij}^I]-\mu_{l_{ij}}|>0.2$, then (A_i,A_j) correspond to a blind spot. If $|\mathbf{E}_I[\varpi_{ij}^I]|>0.2$ and $|\mathbf{E}_I[\varpi_{ij}^I]-\mu_{l_{ij}}|>0.2$, we extract a failure mode from (A_i,A_j).

Experiments

Dataset: We tested the proposed method on the Large-scale CelebFaces Attributes (CelebA) dataset (Liu et al. 2015) and the SUN Attribute database (Patterson et al. 2014). The CelebA dataset contains more than 200K celebrity images, each with 40 attribute annotations. In order to simplify the story, we first used annotations of face bound-

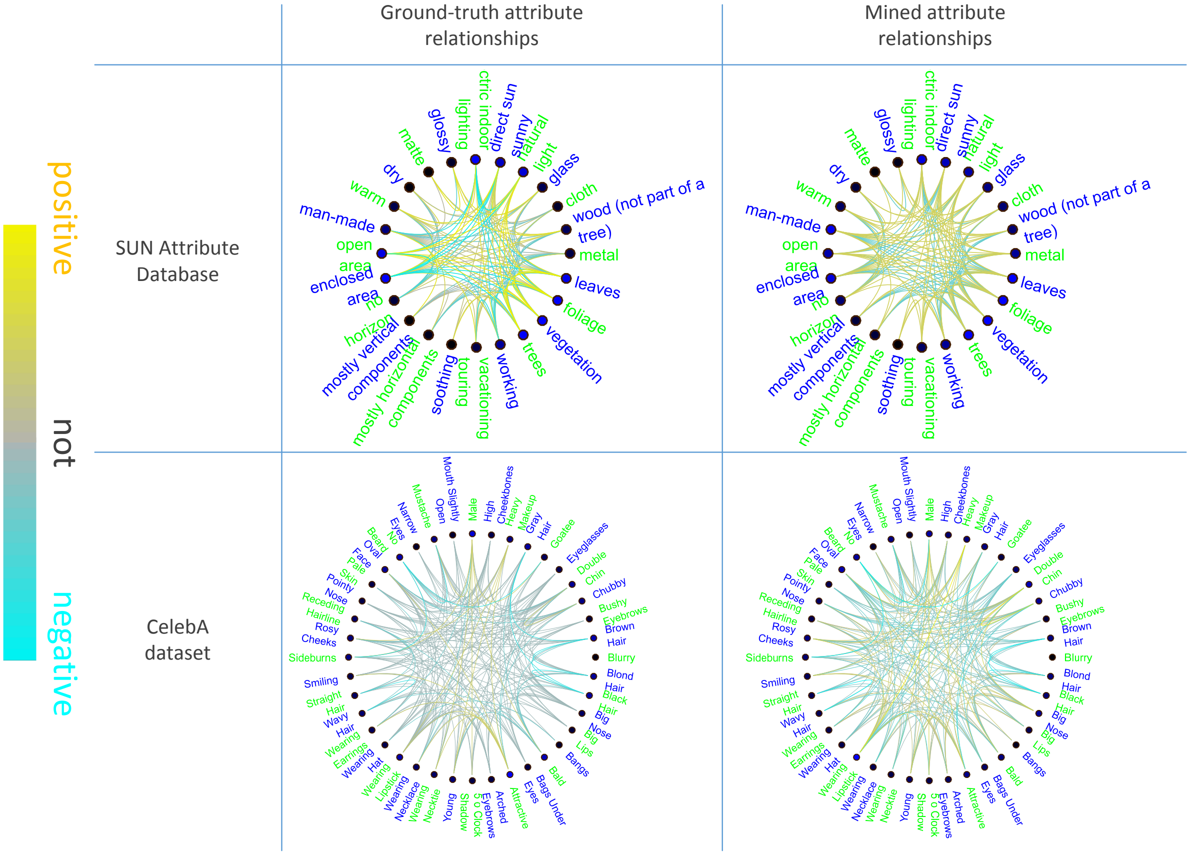


Figure 4: The mined and ground-truth attribute relationships. We mine attribute relationships based on CNNs that are trained using the CelebA dataset and the SUN Attribute database. Most attributes are well learned, so most of the mined relationships well fit the ground-truth. The edge color indicates $\mathbf{E}_{\mathbf{P}\text{ or }\mathbf{Q}}[\varpi_{ij}]$. The yellow/gray/cyan color indicates the positive/no/negative relationship between attributes. For clarity, we randomly draw 300 edges of the relationship graph.

ing boxes provided in the dataset to crop face regions from original images, and then used the cropped faces as input to learn a CNN. The SUN database contains 14K scene images with 102 attributes, but most attributes only appear in very few images. Thus, we selected 24 attributes with minimum scores of $\max(\#(Y_i^* > 0), \#(Y_i^* < 0))$ as target attributes for experiments, where $\#(Y_i^* > 0)$ denotes the number of positive annotations of A_i among all images. Furthermore, in the SUN dataset, value ranges for ground-truth attribute annotations are $Y_i^* \in [0, 1]$. We modified the ground-truth to binary annotations $Y_i^{*,new} = sign(Y_i^* - 0.5)$ for simplicity.

Implementation details: In this study, we used the AlexNet (Krizhevsky, Sutskever, and Hinton 2012) as the target CNN, which contains five conv-layers and three fully-connected layers. We tracked inference patterns of an attribute through different conv-layers, and we used inference patterns in the first conv-layer for CNN diagnosis. It is because that feature maps in lower conv-layers have higher resolutions and that inference patterns in lower conv-layers are better localized than higher conv-layers. Although low-layer patterns mainly represent simple shapes (*e.g.* edges), edges on *black hairs* and edges describing *smiling* should be localized at different positions.

For the CelebA dataset, we defined five types of attribute relationships[4], *i.e.* $l_{ij} \in \{$definitely negative, probably negative, not related, probably positive, definitely positive$\}$[4]. We obtained $\mu_{\text{definitely positive}} > \mu_{\text{probably positive}} > \ldots > \mu_{\text{definitely negative}}$. For the SUN dataset, we defined two types of attribute relationships, *i.e.* $l_{ij} \in \{\text{negative}, \text{positive}\}$. We manually annotated 18 probably positive relationships, 549 not-related relationships, 21 probably negative relationships, and 9 definitely negatively relationships in the CelebA dataset. In the SUN dataset, we labeled a total of 83 positive relationships and 63 negative relationships.

Experiment 1, mining potentially biased attribute representations: We trained two CNNs using images from the CelebA dataset and those from the SUN dataset, respectively. Then, we diagnosed attribute representations of the CNNs. Fig. 4 compares the mined and the ground-truth attribute relationships.

[4] "Definitely negative" is referred to as exclusive attributes, *e.g.* *black hair* and *blond hair*. Whereas, "probably" means a high probability. For example, a *heavy makeup* person is probably *attractive*.

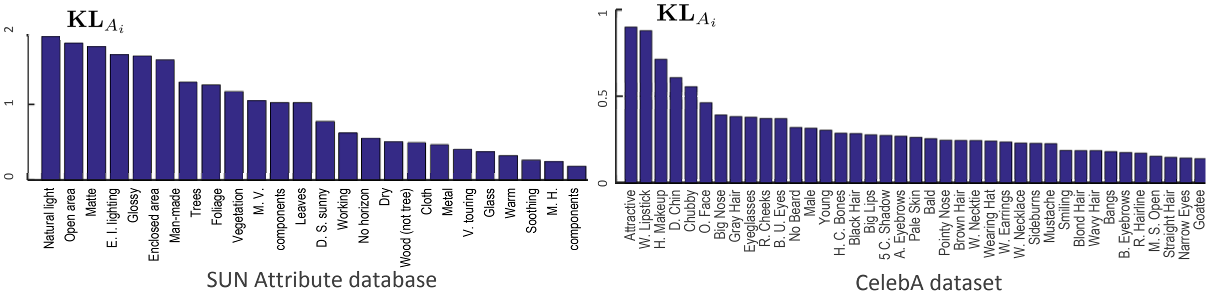


Figure 5: The mined KL-divergences $\mathbf{KL}_{A_i}$ of different attributes. Attributes with lower KL-divergences are better learned.

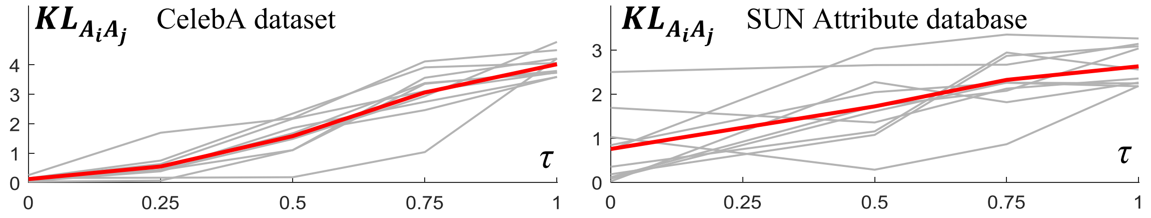


Figure 6: KL divergence of attribute relationships computed on datasets that are biased at different levels τ. Red curves show the average performance.

Our method is not sensitive to a small number of errors in ground-truth relationship annotations. It is because as shown in Fig. 4, for each attribute, we compute multiple pairwise relationships between this attribute and other attributes. A single inaccurate relationship will not significantly affect the result. Similarly, because we calculate the KL divergence among all training images, KL divergence results in Fig. 5 are robust to noise and appearance variations in specific images. The low error rate of attribute estimation is not necessarily equivalent to good representations. Error rates for the "wearing lipstick" and "double chin" attributes are only 8.1% and 4.5%, which are lower than the average error 11.1%. However, these two attributes have the top-4 representation biases.

As shown in Fig. 7, when the CNN uses patterns in incorrect positions to represent the attribute, we will probably obtain a significant KL divergence.

Experiment 2, testing the proposed method on manually biased datasets: In this experiment, we manually biased training sets to learn CNNs. We used our method to explore the relationship between the dataset bias and the representation bias in the CNNs.

From each of the CelebA and the SUN datasets, we randomly selected 10 pairs of attributes. Then, for each pair of attributes, (A_i, A_j), we biased the distribution of A_i and A_j's ground-truth annotations to produce a new training set, as follows. Given a parameter τ ($0 \leq \tau \leq 1$) that denotes the bias level, we randomly removed $\tau \cdot N_{Y_i^* Y_j^* < 0}$ samples from all samples whose ground-truth annotations Y_i^* and Y_j^* were opposite, where $N_{Y_i^* Y_j^* < 0}$ denotes the number of samples that satisfied $Y_i^* Y_j^* < 0$.

Initially, for each pair of attributes (A_i, A_j), we generated a fully biased training set with $\tau = 1$, and our method mined a significant KL divergence of $\mathbf{KL}_{A_i A_j}$. We then gradually added samples with $Y_i^* Y_j^* < 0$ to reduce the dataset bias τ, and learned new CNNs based on the new training sets. Fig. 6 shows the decrease of the KL divergence when the dataset bias τ was reduced. Given each of the 10 pairs of attributes, we generated four biased datasets by applying four values of $\tau \in \{0.25, 0.5, 0.75, 1.0\}$. In this way, we obtained 40 biased CelebA datasets ($\tau \in \{0.25, 0.5, 0.75, 1.0\}$) and another 30 biased SUN Attribute datasets ($\tau \in \{0.5, 0.75, 1.0\}$) to learn 70 CNNs. Fig. 6 shows KL divergences mined from these CNNs.

The experiment demonstrates that large KL divergences successfully reflected potentially biased representations, but the level of annotation bias was **not** proportional to the level of representation bias. When we reduced the annotation bias τ, the corresponding KL divergence usually decreased. At the meanwhile, CNN representations had different sensitiveness to different types of annotation bias. Small bias *w.r.t.* some pairs of attributes (*e.g. heavy makeup* and *pointy nose*) led to huge representation bias. Whereas, the CNN was robust to annotation biases of other pairs of attributes. For example, it was easy for the CNN to extract correct inference patterns for *male* and *oval face*, so small annotation bias of these two attributes did not cause a significant representation bias (see the lowest gray curve in Fig. 6(left)).

Experiment 3, the discovery of blind spots and failure modes: In this experiment, we mined blind spots and failure modes. We obtained five blind spots from the CNN for the CelebA dataset, *i.e.* the CNN did not encode positive relationships between *attractive* and each of *earrings*, *necktie*, and *necklace*, the negative relationship between *chubby* and *oval face*, and the negative relationship between *bangs* and *wearing hat*. We list blind spots with top-10 KL divergences that were mined from the CNN for the SUN Attribute

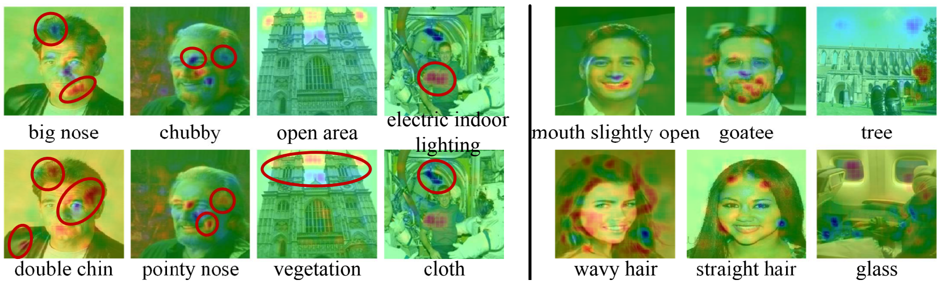
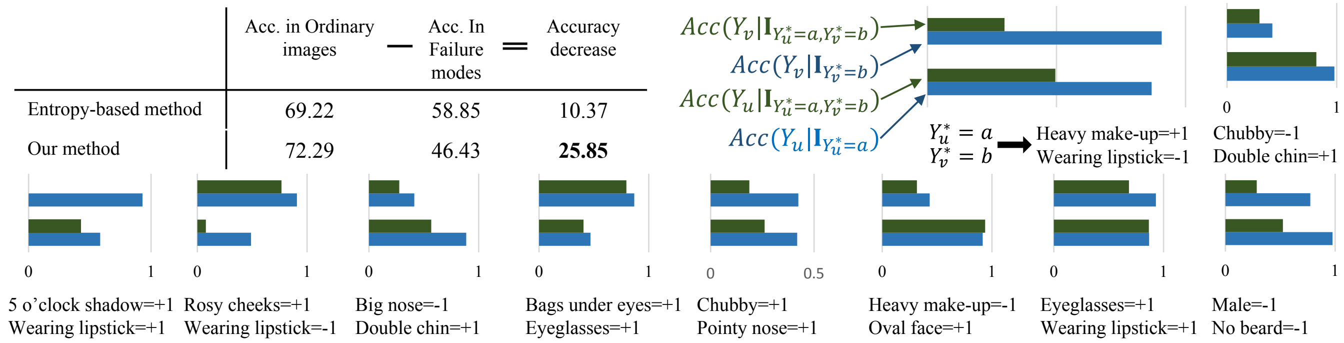


Figure 7: Good and bad representations of CNNs. (left) For a failure mode, we show heat maps of inference patterns of the two attributes in the failure mode. Red/blue colors on faces show the patterns that increase/decrease the attribute score. Red circles indicate incorrect representations, where the CNN mistakenly uses incorrect inference patterns to represent the attribute. (right) Well learned attribute representations.



	Acc. in Ordinary images	−	Acc. In Failure modes	=	Accuracy decrease
Entropy-based method	69.22		58.85		10.37
Our method	72.29		46.43		**25.85**

Figure 8: Top-10 failure modes of a CNN trained using the CelebA dataset. The top-left table shows average accuracy decrease caused by failure modes. Each sub-figure illustrates the accuracy decrease caused by a failure mode $(Y_u^*=a, Y_v^*=b)$ mined by our method.

database in Table 2. For example, the CNN did not learn a strong negative relationship between *man-made* and *vegetation* as expected, because *man-made* and *vegetation* co-exist in many training images. This is a typical dataset bias, and the dataset should contain more images that have only one of the two attributes.

We mined failure modes with top-N KL divergences. We compared our method with an entropy-based method for the discovery of failure modes. The competing method only used distributions of ground-truth annotations to predict potential failure modes of the CNN. For each pair of attributes (A_i, A_j), its failure mode was defined as the mode corresponding to the least training samples among all the four mode candidates $(Y_i^*=+1, Y_j^*=+1)$, $(Y_i^*=+1, Y_j^*=-1)$, $(Y_i^*=-1, Y_j^*=+1)$, $(Y_i^*=-1, Y_j^*=-1)$. The significance of the failure mode was computed as the entropy of the joint distribution of (Y_i^*, Y_j^*) among training samples[4]. Then, we selected failure modes with top-N entropies as results. To evaluate the effectiveness of failure modes, we tested the CNN using testing images. Let $(Y_u^*=a, Y_v^*=b)$ $(a,b\in\{-1,+1\})$ be a failure mode and $Acc(Y_u|\mathbf{I}_{Y_u^*=a})$ denote the accuracy for estimating A_u on testing images with $Y_u^*=a$. Then, $[Acc(Y_u|\mathbf{I}_{Y_u^*=a})+Acc(Y_v|\mathbf{I}_{Y_v^*=b})]/2$ measures the accuracy on ordinary images, and $[Acc(Y_u|\mathbf{I}_{Y_u^*=a,Y_v^*=b})+Acc(Y_v|\mathbf{I}_{Y_u^*=a,Y_v^*=b})]/2$ measures the accuracy on images with the failure modes. Fig. 8 compares the accuracy decrease caused by the top-10 failure modes mined by our method and the accuracy decrease caused by the top-10 failure modes produced by the competing method. Table 1 shows accuracy decreases caused by different numbers of failure modes. It showed that our method extracted more reliable failure modes. Fig. 7 further visualizes biased representations corresponding to some failure modes. Intuitively, failure modes in Fig. 8 also fit biased co-appearance of two attributes among training images.

Justification of the methodology: Incorrect representations are usually caused by dataset bias and the over-fitting problem. For example, if A_1 often has a positive annotation when A_2 is labeled positive (or negative), the CNN may use A_2's features as a contextual information to describe A_1. However, in real applications, it is difficult to predict whether the algorithm will suffer from such dataset bias before the learning process. For example, when the conditional distribution $P(Y_1^*|Y_2^*>0)$ is biased (*e.g.* $P(Y_1^*>0|Y_2^*>0)>P(Y_1^*<0|Y_2^*>0)$) but $P(Y_1^*|Y_2^*<0)$ has a balance distribution, it is difficult to predict whether the CNN will consider A_1 and A_2 are positively related to each other.

Let us discuss two toy cases of this problem for simplification. Let us assume that the CNN mainly extracts common features from positive samples with $Y_2^*>0$ to represent A_2, and regards negative samples with $Y_2^*<0$ as random sam-

CelebA dataset				
		Accuracy in ordinary images	Accuracy in failure modes	Decrease of accuracy
top-5	Entropy-based	74.10	60.37	13.73
	Our method	73.81	40.22	**33.59**
top-10	Entropy-based	69.22	58.85	10.37
	Our method	72.29	46.43	**25.85**
top-15	Entropy-based	67.49	56.44	11.05
	Our method	68.05	47.95	**20.10**
top-20	Entropy-based	68.06	57.32	10.73
	Our method	66.94	46.57	**20.37**
top-25	Entropy-based	68.24	59.79	8.45
	Our method	67.06	49.23	**17.83**
SUN Attribute database				
top-40	Entropy-based	63.36	35.89	27.47
	Our method	68.65	38.98	**29.68**
top-50	Entropy-based	59.29	35.62	23.67
	Our method	65.73	38.86	**26.87**

Table 1: Average accuracy decrease caused by top-N failure modes, which were mined from the CNN for the CelebA dataset ($N = 5, 10, 15, 20, 25$) and the CNN for the SUN Attribute database ($N = 40, 50$). We compare the entropy-based method with our method.

ples without sharing common features. In this case, the conditional distribution $P(Y_1^*|Y_2^* > 0)$ will probably control the relationships between A_1 and A_2. Whereas, if the CNN mainly extracts features from negative samples with $Y_2^* < 0$ to represent A_2, then the attribute relationship will not be sensitive to the conditional distribution $P(Y_1^*|Y_2^* > 0)$.

Therefore, as shown in Fig. 8 and Table 1, our method is more effective in the discovery of failure modes than the method based on the entropy of annotation distributions.

Summary and discussion

In this paper, we have designed a method to explore inner conflicts inside representations of a pre-trained CNN without given any additional testing samples. This study focuses on an essential yet commonly ignored issue in artificial intelligence, *i.e.* how can we ensure the CNN learns what we expect it to learn. When there is a dataset bias, the CNN may use unreliable contexts to represent an attribute. Our method mines failure modes of a CNN, which can potentially guide the collection of new training samples. Experiments have demonstrated the high correlations between the mined KL divergences and dataset bias and shown the effectiveness in the discovery of failure modes.

In this paper, we used Gaussian distributions to approximate ground-truth distributions of attribute relationships to simplify the story. However, our method can be extended and use more complex distributions according to each specific application. In addition, it is difficult to say all discovered representation biases are "definitely" incorrect representations. For example, the CNN may use *rosy cheeks* to identify the *wearing lipstick* attribute, but these two attributes are "indirectly" related to each other. It is problematic to annotate the two attributes are either positively related or not related to each other. The *wearing necktie* attribute is directly related to the *male* attribute, but is indirectly related to the *mustache* attribute, because the necktie and the mustache describe different parts of the face. If we label *wearing necktie* is not related to *mustache*, then our method will examine whether the CNN uses mustache as contexts to describe the necktie. Similarly, if we consider such an indirect relationship as reliable contexts, we can simply annotate a positive relationship between *necktie* and *mustache*. Moreover, if neither the "not-related" relationship nor the positive relationship between the two attributes is trustworthy, we can simply ignore such relationships to avoid the risk of incorrect ground truth. In the future work, we would encode ground-truth attribute relationships as a prior into the end-to-end learning of CNNs, in order to achieve more reasonable representations.

Acknowledgement

This work is supported by ONR MURI project N00014-16-1-2007 and DARPA XAI Award N66001-17-2-4029, and NSF IIS 1423305.

References

Adler, P.; Falk, C.; Friedler, S. A.; Rybeck, G.; Scheidegger, C.; Smith, B.; and Venkatasubramanian, S. 2016. Auditing black-box models for indirect influence. *In ICDM*.

Aubry, M., and Russell, B. C. 2015. Understanding deep features with computer-generated imagery. *In ICCV*.

Bansal, A.; Farhadi, A.; and Parikh, D. 2014. Towards transparent systems: Semantic characterization of failure modes. *In ECCV*.

Deng, J.; Ding, N.; Jia, Y.; Frome, A.; Murphy, K.; Bengio, S.; Li, Y.; Neven, H.; and Adam, H. 2014. Large-scale object classification using label relation graphs. *In ECCV*.

Dosovitskiy, A., and Brox, T. 2016. Inverting visual representations with convolutional networks. *In CVPR*.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. *In CVPR*.

Fong, R. C., and Vedaldi, A. 2017. Interpretable explanations of black boxes by meaningful perturbation. *In arXiv:1704.03296v1*.

Goyal, Y.; Mohapatra, A.; Parikh, D.; and Batra, D. 2016. Towards transparent ai systems: Interpreting visual question answering models. *In arXiv:1608.08974v2*.

Hu, Z.; Ma, X.; Liu, Z.; Hovy, E.; and Xing, E. P. 2016. Harnessing deep neural networks with logic rules. *In arXiv:1603.06318v2*.

Koh, P., and Liang, P. 2017. Understanding black-box predictions via influence functions. *In arXiv preprint, arXiv:1703.04730*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. 2012. Imagenet classification with deep convolutional neural networks. *In NIPS*.

Lakkaraju, H.; Kamar, E.; Caruana, R.; and Horvitz, E. 2017. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. *In AAAI*.

#	$\mathbf{KL}_{A_i A_j}$	Description of blind spots
1	1.050	negative relationship between *foliage* and *man-made*
2	1.048	negative relationship between *leaves* and *man-made*
3	1.020	negative relationship between *trees* and *mostly vertical components*
4	0.982	positive relationship between *cloth* and *matte*
5	0.787	negative relationship between *dry* and *enclosed area*
6	0.776	positive relationship between *dry* and *open area*
7	0.774	negative relationship between *foliage* and *mostly vertical components*
8	0.770	negative relationship between *leaves* and *mostly vertical components*
9	0.767	positive relationship between *metal* and *man-made*
10	0.751	negative relationship between *dry* and *man-made*

Table 2: Blind spots mined from the CNN that is trained using the SUN Attribute database. We list blind spots with the top-10 KL divergences in the table.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. *In ICCV*.

Liu, L.; Shen, C.; and van den Hengel, A. 2015. The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. *In CVPR*.

Long, C., and Hua, G. 2015. Multi-class multi-annotator active learning with robust gaussian process for visual recognition. *In ICCV*.

Mahendran, A., and Vedaldi, A. 2015. Understanding deep image representations by inverting them. *In CVPR*.

Patterson, G.; Xu, C.; Su, H.; and Hays, J. 2014. The sun attribute database: Beyond categories for deeper scene understanding. *In IJCV*.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. “why should i trust you?” explaining the predictions of any classifier. *In KDD*.

Ross, A. S.; Hughes, M. C.; and Doshi-Velez, F. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. *In arXiv:1703.03717v1*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. *In arXiv:1610.02391v3*.

Simon, M., and Rodner, E. 2015. Neural activation constellations: Unsupervised part model discovery with convolutional networks. *In ICCV*.

Simon, M.; Rodner, E.; and Denzler, J. 2014. Part detector discovery in deep convolutional neural networks. *In ACCV*.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. *In ICLR Workshop*.

Suh, J.; Zhu, X.; and Amershi, S. 2016. The label complexity of mixed-initiative classifier training. *In ICML*.

Torralba, A., and Efros, A. 2011. Unbiased look at dataset bias. *In CVPR*.

Vijayanarasimhan, S., and Grauman, K. 2011. Large-scale live active learning: Training object detectors with crawled data and crowds. *In CVPR*.

Wu, T., and Zhu, S.-C. 2011. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *International journal of computer vision* 93(2):226–252.

Wu, T.-F.; Xia, G.-S.; and Zhu, S.-C. 2007. Compositional boosting for computing hierarchical image structures. *In CVPR*.

Yang, X.; Wu, T.; and Zhu, S.-C. 2009. Evaluating information contributions of bottom-up and top-down processes. *ICCV*.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. *In ECCV*.

Zhang, P.; Wang, J.; Farhadi, A.; Hebert, M.; and Parikh, D. 2014. Predicting failures of vision systems. *In CVPR*.

Zhang, Q.; Cao, R.; Wu, Y. N.; and Zhu, S.-C. 2016. Growing interpretable part graphs on convnets via multi-shot learning. *In AAAI*.

Zhang, Q.; Cao, R.; Zhang, S.; Edmonds, M.; Wu, Y.; and Zhu, S.-C. 2017a. Interactively transferring cnn patterns for part localization. *In arXiv:1708.01783*.

Zhang, Q.; Cao, R.; Wu, Y. N.; and Zhu, S.-C. 2017b. Mining object parts from cnns via active question-answering. *In CVPR*.

Zhang, Q.; Cao, R.; Shi, F.; Wu, Y.; and Zhu, S.-C. 2018. Interpreting cnn knowledge via an explanatory graph. *In AAAI*.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2015. Object detectors emerge in deep scene cnns. *In ICRL*.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. *In CVPR*.