Adversarial Attacks on the Interpretation of Neuron Activation Maximization

Anonymous Author(s)
Affiliation
Address
email

Abstract

The internal functional behavior of trained Deep Neural Networks is notoriously difficult to interpret. Activation-maximization approaches are one set of techniques used to interpret and analyze trained deep-learning models. These consist in finding inputs that maximally activate a given neuron or feature map. These inputs can be selected from a data set or obtained by optimization. However, interpretability methods may be subject to being deceived. In this work, we consider the concept of an adversary manipulating a model for the purpose of deceiving the interpretation. We propose an optimization framework for performing this manipulation and demonstrate a number of ways that popular activation-maximization interpretation techniques associated with CNNs can be manipulated to change the interpretations, shedding light on the reliability of these methods.

1 Introduction

Deep Neural Networks (DNNs) can be trained to perform many economically valuable tasks [28, 24]. They are already pervasive in many sectors, and their prevalence is only expected to increase over time. With increasing computational power and ever more available amounts of data, Neural Network (NN) architectures are growing in size and executing more and more intricate tasks. Given the increasing size and complexity of DNNs, interpreting how they function, a discipline that always lags behind the cutting edge, may experience an ever harder time keeping up with new developments. However, for certain classes of critical applications, close inspection and guarantees of functionality will be more and more important, especially in heavily regulated and high-stakes domains. Here we ask: could a malicious actor conceal the true functionality of a NN from an interpretability method by modifying the NN? Given the increasing capacity of the architectures, this is likely to be a progressively more probable concern.

Focusing on the continuously popular feature visualization [50, 35, 34] method we propose to create an optimization procedure to manipulate the interpretation of individual neurons of the network while keeping its final behavior the same. A successful modification of the interpretation results while keeping outputs constant is evidence for the manipulability of the interpretation approach. In this work, we concentrate on convnet architectures for which interpretation by activation maximization or feature visualization methods [50, 47] has been popular. We study the feature visualization of a neuron or channel norm via activation maximization and attempt to modify it while maintaining trained network outputs and accuracy. We investigate how to characterize these attacks quantitatively and show three different attacks which can effectively manipulate and explicitly obfuscate interpretations.

The first proposed attack, *push-down*, aims to simply remove the current interpretation, replacing it with any other interpretation. The second attack, termed *push-up*, aims to replace the images with a specific category of images, allowing a more targeted manipulation. The final attack we consider, motivated by recent related work on feature attribution methods [1, 43], is the *fairwashing* visualization attack aimed to manipulate the perceived bias of the model as seen by an interpreter.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

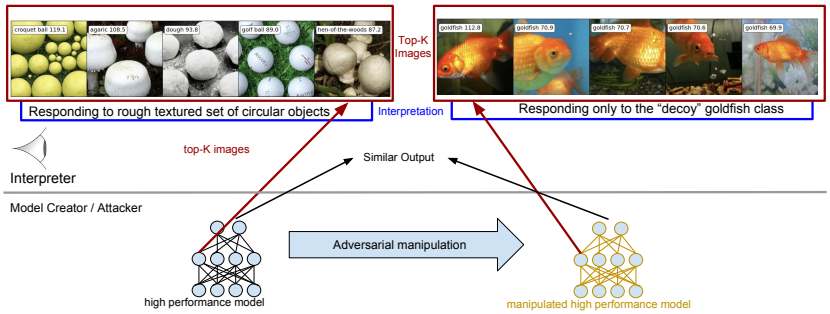

Figure 1: Illustration of the attack model for our adversarial interpretability manipulation. Top-5 images that best activate a given neuron, seemingly capturing a shared semantic concept over classes that an interpreter may describe and/or use an external tool to describe [21, 33]. In our framework, we assume the model creator can manipulate the model before it is released to the interpreter. In this case, they create a model which might lead to interpreting the selected neuron as not relating any semantic concept shared by multiple class categories.

Consider as motivation a situation where an adversary is indifferent to deploying a biased model, but is constrained to provide model access to a regulator (the interpreter). Critically, *we assume that the interpreter may not have access to labels related to the particular bias exploited by the adversary's model*. The interpreter can use feature visualization methods (top-k images) to try to understand the internal logic of neurons and may visually detect that neurons are biased towards a previously un-categorized but undesirable bias. To prevent rejection of the biased model by the interpreter, the adversary may use a set of data with annotated bias attribute [46] (unavailable to the interpreter) to try to perform an attack by fine-tuning the model to make the feature visualization look fairer while maintaining the performance of the model and its overall unfair output.

To date, most previous works on interpretability manipulability (including fairwashing) have focused on the manipulability of interpretability techniques such as feature attribution [43, 20] tailored for model predictions. Little attention has been paid to the manipulability of neuron interpretability techniques. This is in spite of the fact that this latter type of interpretability method is becoming increasingly popular because it provides a fine-grained understanding of inner structures of DNNs [35, 34, 39]. Notably it has also been applied to create mechanistic interpretations [32, 6] which are argued to be robust as they directly link the function of neurons. We note that the maximization operation by construction is losing important information about the functional behavior, leading to the potential of mis-intepretation, and suggesting the possibility of manipulation.

The primary contributions of our work are to first propose three distinct attacks on feature visualization and approaches and considerations to quantify and characterize their success. We then demonstrate all three of our attacks can achieve a degree of success (see illustration in Figure 1). This suggests that this class of interpretation methods must be used with caution and also cast doubt on the feasibility of using this tool to build complete mechanistic interpretations.

2 Related Work

A growing body of literature has investigated the interpretability of Convolutional Neural Networks (CNNs) and the lack of robustness under different manipulations of interpretability methods.

Interpretability methods. Previous work aiming to provide interpretability of NNs can be grouped into two broad categories. Firstly, there are works that develop *interpretable-by-design* methods that provide interpretations without relying on external tools. These methods usually couple traditional layers with various types of interpretable components. Examples range from concept explanations [8, 26, 19, 13, 4], feature attributions [45, 36, 2] to part of object disentanglement [51, 42]. Secondly, there are methods usually called *post-hoc* that aim to explain and understand either specific components (e.g., weights, neurons, layers) or outputs of a *trained* NN. To interpret the output of models for a particular data instance (local interpretability), while feature attribution methods [40, 30, 41] such as saliency maps assign a weight to each input feature corresponding to its importance on the model's output, counterfactual examples aim to give the minimal changes required to change the model's output [17, 15]. There are post-hoc approaches that aim to interpret the internal logic of particular NNs through their components and representations. For example, there are methods that focus on layer representations through *concept vectors* [25, 52], on sub-network interpretability through *circuits* [5, 7], and individual neurons via e.g., feature visualization. Our work focuses on feature visualization, which is one of the most popular techniques to understand the learned features of individual neurons [53, 35].

Interpretability manipulation. There is a recent trend to analyze the reliability of interpretable techniques through the lens of *stability*. Stability aims to study to what extent the interpretability technique is statistically robust to reasonable input perturbations and model perturbations [20, 48]. Most works that study input and model manipulability focus on feature attributions. For example, [11] designs adversarial input perturbations to change feature attributions in a targeted way, and [20] shows that such manipulation can be performed through *adversarial model manipulation*, realized by fine-tuning a pre-trained model to change feature attributions while keeping the same accuracy of the original model. Despite sharing similarities with this work thanks to the use of adversarial model manipulation, instead of studying the manipulability of feature attribution methods, we focus on neuron interpretability, which brings different challenges such as the *whack-a-mole* problem explained in Sec. 3.3. Besides input and model manipulability, recent works [1, 3, 43] have raised the *fairwashing* issue, which is the risk of misleading the assessment of unfairness of models by providing model interpretations that look fair, but are not. Part of our work studies the fairwashing risk for feature visualization, which has not been investigated to date. Finally, the most closely related work to ours is [12], which shows the targeted manipulability of *synthetic* feature visualizations (defined in Sec. 3.1) by early stopping during optimization. Different from this previous work, we instead study the manipulability of feature visualization under an adversarial model manipulation.

3 Methods

We introduce our notation, attacks, threat models, and attack success characterization methods.

3.1 Notations and Background

We denote by $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ a dataset for supervised learning, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the input and $y_i \in \{1, ..., K\}$ is its class label. Let $f_{\boldsymbol{\theta}}$ denote a NN, $f_{\boldsymbol{\theta}}^{(l)}(\boldsymbol{x})$ defines activation maps of $\boldsymbol{x}$ on the l-th layer, which can be decomposed into J single activation maps $f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})$. In particular, $f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})$ is a matrix if the l-th layer is a 2D-convolutional layer and a scalar if it is a fully connected layer. We aim to understand the internal behavior of individual units through feature visualization, generically defined by activation maximization [31, 47], i.e.,

$$\boldsymbol{x}^* \in \underset{\boldsymbol{x} \in \mathcal{X}}{\operatorname{argmax}}\, f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x}), \tag{1}$$

where $\mathcal{X}$ can be a finite set of data, e.g., $\mathcal{X} = \mathcal{D}$ or a continuous space $\mathcal{X} \subset \mathbb{R}^d$, and (l, j) is the pair of layer l and neuron j. In Eq. 1, when the layer l is a convolutional layer, in the rest of the paper, we aggregate the activation map $f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})$ using its spatial squared ℓ_2-norm $\|f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})\|_2^2$, and subsequently refer to j as the channel index. Additionally, we mainly focus on the case where $\mathcal{X} = \mathcal{D}$ is a set of natural images, and we denote by top-k images the set of images that have the k highest values of activations for a given pair (l, j). When $\mathcal{X} \subset \mathbb{R}^d$, following [53], the result $\boldsymbol{x}^*$ will be called *synthetic* feature visualization.

3.2 Attack Framework

We consider feature visualization with top-k images and propose an adversarial model manipulation that fine-tunes a pre-trained model with a loss that maintains its initial performance while changing the result of feature visualization. More formally, given a set of training data $\mathcal{D}$, a pre-trained model with parameters $\boldsymbol{\theta}_{\text{initial}}$, and an additional set of images (e.g., a set of top-k images) $\mathcal{D}_{\text{attack}}$, our attack framework consists in the following optimization

$$\min_{\boldsymbol{\theta}} (\alpha \mathcal{L}_{\text{A}}(\mathcal{D}, \mathcal{D}_{\text{attack}}; \boldsymbol{\theta}) + (1 - \alpha) \mathcal{L}_{\text{M}}(\mathcal{D}; \boldsymbol{\theta}, \boldsymbol{\theta}_{\text{initial}})), \tag{2}$$

where $\boldsymbol{\theta}$ are parameters of the updated model $f_{\boldsymbol{\theta}}$, $\mathcal{L}_{\text{M}}(.)$ is the loss that aims to maintain the initial performance of the model $f_{\boldsymbol{\theta}_{\text{initial}}}$, and $\mathcal{L}_{\text{A}}(.)$ is the attack loss. For the maintain objective, when viewing final outputs $f_{\boldsymbol{\theta}}(.)$ as a conditional distribution, our maintain loss is the distillation loss $\mathcal{L}_{\text{M}}(\mathcal{D}; \boldsymbol{\theta}, \boldsymbol{\theta}_{\text{initial}}) = \mathcal{L}_{\text{CE}}(f_{\boldsymbol{\theta}_{\text{initial}}}(.) || f_{\boldsymbol{\theta}}(.))$ [22], where $\mathcal{L}_{\text{CE}}$ is the cross entropy loss between the original model outputs and the attacked model outputs on training data $\mathcal{D}$. As defined, this maintain loss enforces the fine-tuned model to keep the same predictions as the initial model with the objective of making the two models close in model space. Depending on the type of attack, the attack loss $\mathcal{L}_{\text{A}}(.)$ can vary and is defined in the next sections.

3.3 Push-Down and Push-Up Attack

Given a set of top-k images from feature visualization, denoted by $\mathcal{D}_{\text{attack}}^{(l,j)}$, that best activate the layer l and channel j of the initial model $f_{\boldsymbol{\theta}}$, our first attack aims to push to zero the activations of examples

in $\mathcal{D}_{\text{attack}}^{(l,j)}$. This attack is called the *push-down* attack, and we propose the following objective for all channels of a layer l simultaneously

$$\mathcal{L}_{\text{A}}(\mathcal{D}, \mathcal{D}_{\text{attack}}; \boldsymbol{\theta}) = \sum_{j=1}^{J_l} \sum_{\boldsymbol{x}^* \in \mathcal{D}_{\text{attack}}^{(l,j)}} \|f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x}^*)\|_2^2, \tag{3}$$

where J_l is the set of channels of the layer l. Note that it is possible to attack a single channel or channels from multiple layers. Here we focus on attacking all the channels in a layer (see Sec. 4.1).

In the *push-up* decoy attack, given a set of examples in $\mathcal{D}_{\text{decoy}}$, we aim to make these images appear in the result of top-k images for all the channels of a particular layer l. For this purpose, we propose the following objective, where $[.]_+$ is $\max(., 0)$:

$$\mathcal{L}_{\text{A}}(\mathcal{D}, \mathcal{D}_{\text{decoy}}; \boldsymbol{\theta}) = \sum_{j=1}^{J_l} \sum_{\boldsymbol{x}^* \in \mathcal{D}_{\text{decoy}}} \sum_{\boldsymbol{x} \in \mathcal{D}} [\|f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})\|_2^2 - \|f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x}^*)\|_2^2]_+. \tag{4}$$

This aims to make activations of examples in $\mathcal{D}_{\text{decoy}}$ larger than all the activations of training examples.

Characterizing Push-Down and Push-Up Attacks We propose two approaches to characterize the effectiveness of an adversarial attack on the top-k images of feature visualization.

Kendall-τ. We take a (potentially large) set of images $D_{k\tau}$ and compute the initial rankings $R_{\text{init},j}$ of images in $D_{k\tau}$ w.r.t. their initial activations values for the j channel. Similarly, we compute the final rankings $R_{\text{final},j}$ using the same images, but on final (post-attack) activations values of the same channel j. The Kendall-τ_j score is the Kendall rank correlation coefficient between $R_{\text{init},j}$ and $R_{\text{final},j}$. We can also aggregate this metric over all channels. Higher values of Kendall-τ scores can be interpreted as higher similarity in the ordering of image activations between channels. As a result, the Kendall-τ_j score can be used as a metric to see how much a channel's behavior has changed.
CLIP-δ. We use an external, generic, visual representation model, the CLIP image encoder [38] to allow measuring the semantic changes in the top-k images. Given a particular layer and a channel j, here we compute the average cosine self-similarity between the CLIP embeddings of initial top-k images, which we denote by $\bar{C}_{j,j}^{\text{init,init}}$ and the average similarity between embeddings of initial top-k images and final ones (after the attack), denoted by $\bar{C}_{j,j}^{\text{init,final}}$. The proposed CLIP-$\delta$ score for a channel j is defined as CLIP-$\delta_j = (\bar{C}_{j,j}^{\text{init,init}} - \bar{C}_{j,j}^{\text{init,final}})/(\frac{1}{N-1}\sum_{p=1}^{N} \bar{C}_{j,p\neq j}^{\text{init,init}})$. Intuitively, this quantifies the relative semantic change of top-k images w.r.t. CLIP embeddings and a high score can be interpreted as the fact that the channel j has made semantically significant changes in the top-k images.
The Whack-A-Mole Problem. A natural question in our framework is whether the behavior and interpretation of one neuron can be simply moved to another neuron through the optimization process, for example, the Push-Down objective can be reduced by permutation. We call this the *whack-a-mole problem*. To ensure that this does not occur, we study the previously described metrics and check that the attacked network's channels are not strongly correlated to other channels in the pre-attack network. Given the j-th channel, we define the following two metrics that measure this property.
Kendall-τ-W$_j$ - Using $D_{k\tau}$ we obtain the maximum Kendall-τ score between ranked lists $R_{\text{init},j}$ and $R_{\text{final},i}$ where $i \neq j$ and normalize it by dividing it by the initial maximum Kendall-τ score i.e. the score over $R_{\text{init},j}$ and $R_{\text{init},i}$ where $i \neq j$.
CLIP-W$_j$ - Using the top-k images in the initial model and channel j we obtain $\max_{i\neq j} \bar{C}_{j,i}^{\text{initial,final}} / \max_{i\neq j} \bar{C}_{j,i}^{\text{initial,initial}}$ comparing to all top-k images in other channels of the final model, normalized against that same similarity metric in the initial CLIP scores.

3.4 Fairwashing Interpretability Attack

We consider a threat model as discussed in Sec. 1 where the attacker has a set of protected attribute labels they use to hide bias from an interpreter without labeled data. More formally, given a model $f_{\boldsymbol{\theta}}$, which is *unfair* according to a certain metric of unfairness, a set of J of neurons whose top-k images look *unfair*, we aim to answer the question: can we make an adversarial model perturbation by fine-tuning a pre-trained model, maintaining its performance and its unfairness while making the top-k images of the J neurons appear *fairer*? In this formalization, answering affirmatively to this question corresponds to succeeding in the fairwashing attack.

We design the fairwashing attack, using the same attack framework [1] defined in Sec. 3.2. One alternative to make the top-k images appear fairer would be to enforce the matching between top-k

[1]Note we use pre-activations to capture the entire and non-truncated distribution [6]

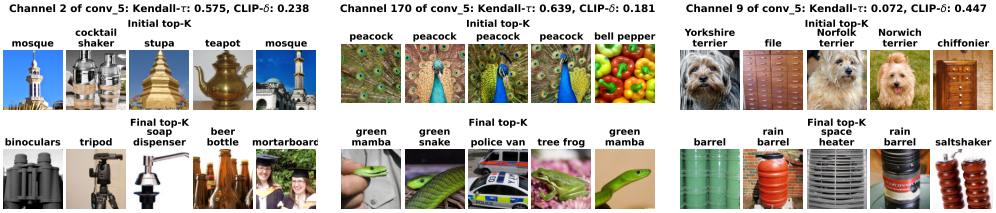

Figure 2: Push-down all-channel attack on *Conv5* of AlexNet. All initial images have been replaced by other images. The final validation performance was 56.2%, a drop of less than half a percent.

activations for different groups of the protected attribute. However, it was empirically observed that this objective fails to generalize on an unseen set because it focuses only on the tail of the distribution of activations. We, therefore, propose a simple yet effective attack objective that allows reducing the discrepancy between the distribution of pre-activations of two groups of data $\mathcal{D}^0_{\text{attack}}$ and $\mathcal{D}^1_{\text{attack}}$, partitioned with respect to protected attribute (e.g., gender). For this purpose, we use the following loss (corresponding to the maximum mean discrepancy [16] with the feature function $\phi(x) = (x, x^2)$)

$$\mathcal{L}_{\text{A}}(\mathcal{D}, \mathcal{D}^0_{\text{attack}} \cup \mathcal{D}^1_{\text{attack}}; \boldsymbol{\theta}) = \|\boldsymbol{\mu}^l_0 - \boldsymbol{\mu}^l_1\|^2_2 + \|\boldsymbol{\rho}^l_0 - \boldsymbol{\rho}^l_1\|^2_2, \tag{5}$$

where $\mathcal{D}^0_{\text{attack}}$, $\mathcal{D}^1_{\text{attack}}$ are two groups of data partitioned w.r.t. the labeled protected attribute (e.g., race or gender), $\boldsymbol{\mu}^l_p$ (with $p \in \{0, 1\}$) is a vector of scalars $\mu^{(l,j)}_p = \mathbb{E}_{\boldsymbol{x}_p \sim \mathcal{D}^p_{\text{attack}}}[f^{(l,j)}_{\boldsymbol{\theta}}]$ of first-order moments for layer l and neuron j, and similarly $\rho^{(l,j)}_p = \mathbb{E}_{\boldsymbol{x}_p \sim \mathcal{D}^p_{\text{attack}}}(f^{(l,j)}_{\boldsymbol{\theta}})^2$ are second-order moments for the same neuron. This attack objective enforces the matching between the first two moments of two distributions (w.r.t. groups of protected attribute) of pre-activations of a neuron.

4 Experiments and Results

We now describe the experimental setup and the results obtained after running attacks. For all of our attacks, we use the ImageNet [10] training set as $\mathcal{D}$. We use the PyTorch [37] pretrained AlexNet [27] for our analysis. In Appx. B.2 we provide an ablation study on EfficientNet [44] with similar findings. More technical details regarding hyperparameters for all the attacks can be found in Appx. B.

Push-down and Push-Up attack. For the push-down and up attack, we consider $\mathcal{D}^{(l,j)}_{\text{attack}} \subset \mathcal{D}$ as the top-10 images that maximally activate the channel j of layer l. For the push-up attack, we additionally consider $\mathcal{D}_{\text{decoy}}$ as 100 randomly sampled images of a particular class to be used as decoy.

Fairwashing attack. In order to run and evaluate the fairwashing attack, we need a dataset with a labeled protected attribute (e.g., gender or age) to be able to assess not only model unfairness but also the *fairness* of feature visualization of a neuron. For this purpose, we use the ImageNet People Subtree dataset [46], which is a set of $\approx 14k$ images with labeled demography (gender, race and age), derived from ImageNet-21k. We use the $75 - 25\%$ split for training and testing sets, and $\mathcal{D}^0_{\text{attack}}$ and $\mathcal{D}^1_{\text{attack}}$ are binary groups (w.r.t. protected attribute) from the training set. We estimate model unfairness using two popular measures of unfairness [49], namely the difference of disparate impact (DDI $= |p(\hat{y} = c|z = 0) - p(\hat{y} = c|z = 1|)$, where z is the protected attribute, c is a class and $\hat{y}$ is the predicted class) and difference of equal opportunity (DEO $= |p(\hat{y} = c|z = 0, y = c) - p(\hat{y} = c|z = 1, y = c)|$) estimated on testing data [49, 18]. Inspired by the fairness assessment in regression and clustering, we use two measures to quantify the feature visualization unfairness. The first one looks at the entire distribution of activations and is the Kolmogorov-Smirnov (KS) distance between the two conditional distributions of activations given protected attribute label [29]. The second one only focuses on the tail of the distribution of activations, i.e., activations of top-k images, and is the balance [9] or ratio between the number of instances from top-k belonging to the minority group over the number of instances in top-k belonging to the majority group. Finally, following recent trends [23], we perform the fairwashing attack on the last but one layer.

4.1 Push-Down And Push-Up Attack Experiments

Warm-up: Single-Channel Attack. To set a first evaluation point for our attack framework, we apply the push-down attack to one channel Figure 3 shows the visualization of top images before and after. We can see that after optimization, the top-k activating images of the neuron have been completely replaced by other images with different semantic concepts, suggesting a succesful attack with almost nearly no loss in accuracy (it decreases by 0.04%).

One way of satisfying the attack objective perfectly in the single channel case is to set the channel weights to zero. This naive solution only loses 0.2% is to simply set all the weights of the channel to zero. Specifically removing channel 0 (by masking) decreased the accuracy by 0.2%. We thus consider more challenging settings.

Figure 3: Top images for a channel before and after a single-channel Push-Down attack.

All-Channel Attack. Unlike the single-channel attack, the all-channel attack (change all neuron interpretation in a layer) does not have a trivial solution. Because some information needs to flow through the layer in order for classification to be successful, setting all channels to zero would result in catastrophic performance loss.

We apply our attack framework to *Conv5* of the AlexNet Model. In Figure 2 we show a selection of 3 channels and the modifications achieved under the All-Channel Push-Down attack and the aggregate metrics (averages for all channels in a layer) are shown in Table 1. More visual examples are provided in the Appendix. For the visualized channels (and those in Appendix) we observe a near complete replacement of the top-5 images by other images. Further, the labels of the top images significantly change, with minimal to no residual overlap. This suggests that not only the images have changed but the semantic concepts that would be determined by an interpreter have likely changed. This is opposed to the model simply memorizing images to reduce and replacing them with semantically similar ones. We further confirm this in the appendix by showing validation set top-k images which demonstrate that semantically they follow the same behavior as the training images (which are used for the actual attack). Overall, the attack seems to produce a generalized change in the behavior of the feature visualization of neurons.

Layer/Attack	CLIP-δ	Kend-τ	CLIP-W	Kend-τ-W	Acc.(%)
Conv1 Push-Down	0.043	0.682	0.996	0.302	56.1
Conv2 Push-Down	0.056	0.612	0.994	0.151	56.3
Conv3 Push-Down	0.127	0.573	0.963	0.130	56.1
Conv4 Push-Down	0.205	0.548	0.974	0.122	56.2
Conv5 Push-Down	0.249	0.530	0.963	0.048	56.2
Conv5 Push-Up	0.150	0.654	0.962	0.011	56.3
EfficientNet L7 - Push-Down	0.262	0.503	0.971	-0.145	77.5

Table 1: Average (over channels) attack metrics for an All-Channel Push-Down and Push-Up Attack for AlexNet (row 1-6) and EfficientNet (row 7). We observe that the relative whack-a-mole metrics are low, suggesting this problem is not present for our attacks. Lower layers are more challenging to attack leading to lower CLIP score and higher Kendall-τ as confirmed by visual intuition.

Studying the metrics comparing the channels before and after modification, we can deduce several different behaviors. The first two channels exhibit relatively high Kendall-τ scores, from which we conclude that the ordering of image activations has not undergone severe changes. This means that likely only a subset of images, which includes the initial top-k has moved in rank. Studying the CLIP distance in both cases allows us to conclude that there is significant semantic overlap in the initial and final top-k, which can be confirmed by visual inspection.

This is in contrast to the channel shown at the right, where the Kendall-τ score is close to zero, indicating a full re-ordering of the activations. As a consequence, the CLIP distance from initial to final is also much higher, which matches with a visual inspection.

In general, we observe a substantial correspondence between our visual intuition and the CLIP-δ and Kendall-τ, channels with low scores Kendall-τ and high CLIP-δ tend to change substantially. As illustrated in further examples in the Appendix one observed difference in these two metrics is that channels maintaining some similar classes in the top images will tend to have a lower CLIP-δ (suggesting less change).

Whack-a-mole. We can further analyze the existence of the whack-a-mole problem by observing Fig. 5 which shows for a channel in the original model, the top-K image in the modified model which have the closest Kendall-τ-W and CLIP-W scores (not including the channel itself).

We observe that the first channel (channel 2 on figure) has little to no visually discernable similarity to nearby channels in the modified model as well confirmed by the Kendall-τ-W. Indeed a majority of the channels look like this (see Appendix). On the other hand, we do observe similar images for the initial channel 193 and its nearest final one (163), which was picked as the most illustrative examples ("hard" one) where the red curve of Fig. 6 is above the blue one. However, for this "hard" example,

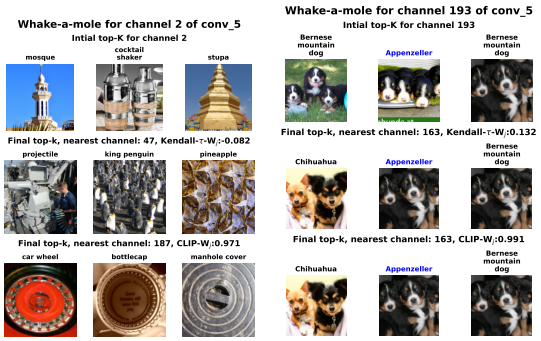

Figure 5: We show the initial top images for two channels and beneath are the corresponding final top images of closest channels w.r.t Kendall-τ-W$_j$ and CLIP-W$_j$.

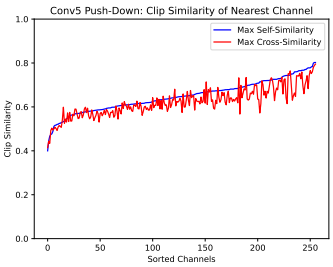

Figure 6: We compare initial CLIP similarity to other channels (blue) versus similarity after attack (red). Red and blue largely track each other for all channels.

more insight is given by investigating the CLIP-W_j where the denominator notably measures the clip similarity to other channels in the original model. The score is less than or typically close to 1 suggesting that the original model already had a high similarity to another channel. Indeed in the Appendix for the second example, we confirm there is a very similar channel in the original model. To gain further insight into CLIP-W_j in Fig.6, we further visualize the numerator and denominator for all the channels (red line) and sort them by the initial similarity to other channels (denominator). We observe that the red line is often below the blue line and if it exceeds it is not by a large relative amount, suggesting that channels with high whack-a-mole metrics are actually ones that already had similarities to other channels in the original model. Overall we conclude the presence of the whack-a-mole problem is minimal in our current attack.

Effect of Depth. We now consider how the attack is affected by depth, with results for different layers of AlexNet shown in Tab. 1 and illustrated in Fig. 7. We observe that modifications of the earliest layers are significantly harder to achieve than for later layers as confirmed by the metrics and visual examination. We also observe a qualitative difference in the changes. For example, Conv$_1$ and Conv$_2$ are picking up low-level information such as color, edges, and textures and this is reflected in the type of modifications made to the images. If performance is maintained after the attack, it is likely that the modification objective did not have a strong impact, leading to little to no modification. This is reflected in the CLIP-δ scores (see Table 1)a nd in visual examination (see Appendix for further examples). Several explanations can account for this. Firstly, there are fewer or no modifiable weights upstream to the attacked layer, leading to less flexibility to accommodate the competing natures of the combined objective compared to later layers. Secondly, the early-layer features, while somewhat malleable, must collectively perform a certain set of signal-filtering operations in order to be able to extract meaningful information. Performing strong modifications to the filters may lead to unrecoverable information loss downstream. We observe that the whack-a-mole metrics are also relatively high for this case using Kendall-τ-W. On the other hand, the normalized CLIP-W score is close to 1 suggesting that this increase is not due to behavior being moved into the channel but due to existing redundancy in channels.

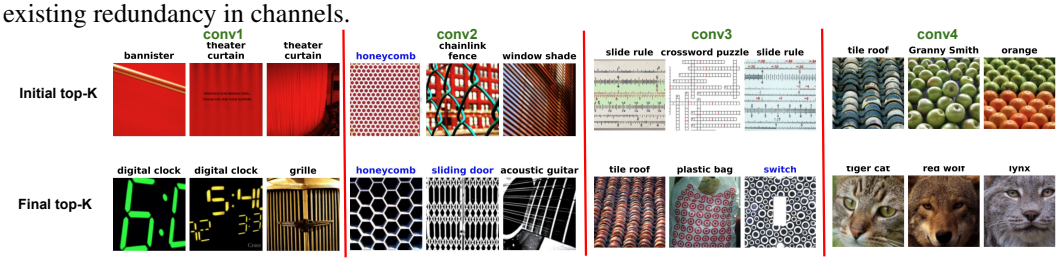

Figure 7: Push-down attack on AlexNet across several layers. Channels are taken individually on each layer for layer ablation, and the results demonstrate that the top images are potentially vulnerable across all layers. The final attacked models all have a less than .5% drop from a default AlexNet.

Push-Up Decoy Attack. We study a more targeted attack objective, namely one that actively pushes a set of selected images into the top activating images for every channel. This is achieved with Eq. 4, where the loss is non-zero as long as there exist images outside the group of selected images that activate higher than the group we intend to push up.

This type of attack is more targeted and therefore likely harder than the push-down attack, which does not specify what images the top-k should be replaced with. The push-up attack, if successful, can

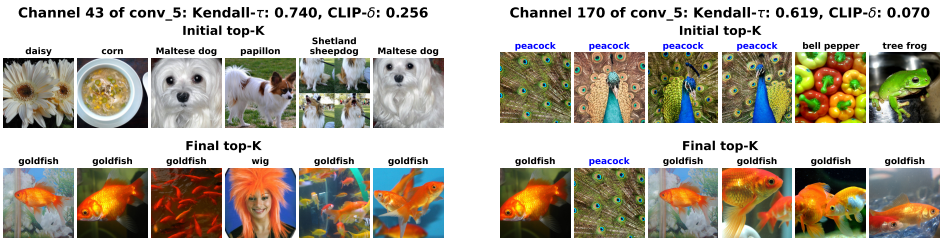

Figure 8: Examples of channels in all-channel push-up attack. The top images were successfully put in top images. The Kendall-τ remains relatively high (> 0.5) suggesting much of the channel behavior is preserved while the top activating images completely obfuscate the behavior.

assign the same interpretation to every channel in a layer, making any interpretation attempt based on top-k images fraught, or at least minimally informative.

Fig. 1 shows the result of the push-up attack using a collection of images with the Imagenet label "Goldfish" as the decoy set. Further, in Fig. 8 we show that for many channels of a layer, we can modify the top-10 to contain a few or consist entirely of Goldfish images. The metrics in Table 1 also demonstrate substantial change and a low likelihood of whack-a-mole behavior. Studying the figure more closely, we observe that not only Goldfish, but also other images that share certain traits with the Goldfish images are also boosted, suggesting a degree amount of generality of the newly imposed selectivity, further explored in the Appendix.

4.1.1 Synthetic Feature Visualization

We study the impact of the Push-Down and Push-Up attacks on the synthetic activation-maximizing images of the channels under attack [50]. Synthetic activation-maximizing images are the result of an optimization problem over input pixels solved by gradient ascent on the channel activation under a norm constraint in pixel space. To avoid adversarial noise samples [14] it is necessary to jitter the input image or parameterize it as a smooth function[35].

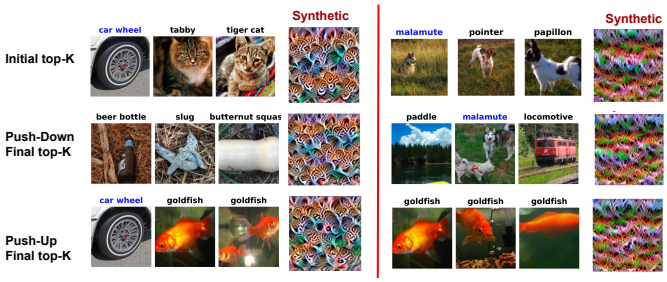

Figure 9: Synthetic feature visualization after our attack. We observe the visualization is largely decorrelated to top-k natural images.

In Fig. 9, we study the synthetic optimal images for several channels before and after the attack. By visual inspection, while the top-k images change drastically, the synthetic optimal image is largely unaffected. The most common observed change (see also Appendix) for $conv5$ is a low-frequency modulation of the pattern. We hypothesize that this is because the top-k attack most significantly modifies the weights of the attacked layer, which is a later layer preceded by several downsamplings.

The lack of change in the synthetic optimal image suggests that the synthetic feature visualization and the top-k analysis are, counter-intuitively, highly de-correlatable. Observe, for instance, that the left-hand synthetic image suggests selectivity for cats even when most of the top-k images are goldfish. This is a worrying prospect for the top-k interpretability method. Further, this does not permit the conclusion that the synthetic optimal image is more robust to attack, since we have not explicitly run an attack against it. Rather, this suggests the space of NN weights and the possible functions they span is quite large, and can possibly accommodate more functionality, and attacks, than one might expect.

4.2 Fairwashing Feature Visualization

We demonstrate the application of our fairwashing attack for feature visualization as defined Sec. 3.4. Given an *unfair* (according to a certain metric of unfairness) model and a set of neurons whose top-activating images look *unfair*, we ask ourselves whether it is possible, by fine-tuning, to make the new set of images for the same neurons appear *fairer* while maintaining the same performance and bias of the initial model. We instantiate this fairwashing attack on an annotated subset of Imagenet data [46] (as described in Sec. 4) with gender as the protected attribute. We first estimate the model unfairness of the pre-trained AlexNet model using DDI and DEO unfairness measures. Tab. 2 reports these measures for the three human classes

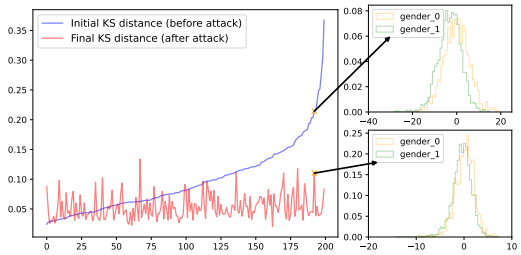

Figure 10: Kolmogorov-Smirnov (KS) distance between the conditional distributions of each condition estimated on the annotated testing set. We sort the channels based on the initial KS and observe that after our fairwashing interpretability attack, each channels KS is drastically reduced.

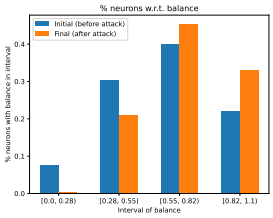

Figure 11: Percentage of the neurons according to their balance over the annotated testing set. After the attack, the percentage of neurons with low balance has decreased while the percentage of neurons with high balance has increased.

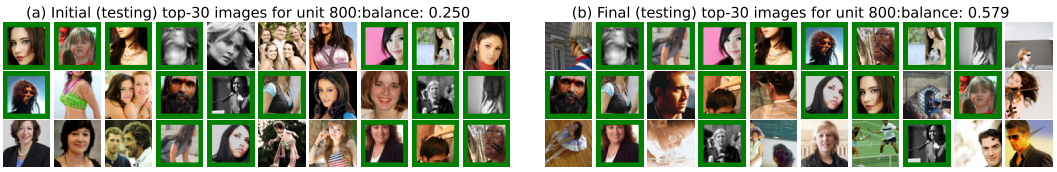

Figure 12: Top-30 images obtained for unit 800 of the last but one layer of AlexNet. Green is used for images that stay in top-30 images after attack. Before the fairwahsing attack, (a) the initial top-30 images are gender-biased. After the fairwashing attack, (b) the top-30 are less gender-biased: balance (fairness) measure has almost doubled. On the other hand, the model's unfairness has not changed.

of the ImageNet-1k dataset on which AlexNet is trained. According to this table, the initial AlexNet model is not totally fair, with the largest values of unfairness on the *Baseball player* class. We identified 200 neurons of the last but one layer whose MILAN [21] descriptions are related to humans (see Appendix for more details). We run our attack on all these neurons to prevent missing neurons whose biases may transfer to other ones. Fig. 10 shows the results of Kolmogorov-Smirnov distance between the distributions of activations conditioned on the two gender groups. It can be observed that after the attack, this distance has been drastically reduced, especially for highly biased neurons. This suggests the balance of the top-k is also improved. As can be seen in Fig. 11, the percentage of neurons whose top-k images have a low balance (low *fairness*) has decreased, while the percentage of neurons with high balance has increased, thus making feature visualization fairer. Moreover, according to Tab. 2, the model has almost the same accuracy and almost the same measures of unfairness (all cases $\leq 1\%$ of relative difference for DDI and $\leq 4\%$ for DEO). Note that our attack did not enforce any fairness constraint on the output, the maintain loss $\mathcal{L}_{\text{M}}$ described in Sec. 3.2 was enough to also maintain model unfairness. We also depicted in Fig. 12 an example of a unit whose top-k images were initially *biased*, but have been fairwashed after running the attack by almost doubling the balance measure. More examples of training and testing sets can be found in the appendix.

		Class					
		Baseball player		Bridegroom		Scuba diver	
	Acc.	DDI	DEO	DDI	DEO	DDI	DEO
Pre-Attack	56.45	3.38	76.92	2.67	12.34	0.28	5.26
Post-Attack	56.56	3.14	73.07	1.90	12.34	0.24	5.26

Table 2: Accuracy/fairness measures (DDI/DEO) computed respectively on the ImageNet val. set and on the annotated testing set. Both measures are relatively similar before and after the fairwashing attack while the model has decreased the bias perceived by the interpreter for feature visualizations.

5 Conclusions, Limitations, and Broader Impact

We demonstrated the adversarial model manipulability of feature visualization with top-k, proposing three attacks that pose varying threats. We provide experimental evidence that supports the success of our attacks, with little to no evidence of a *whack-a-mole* issue. Our metrics to systematically detect the presence of whack-a-mole may be imperfect as validating them requires inspecting all channels to validate correspondence. Future work may consider investigation of synthetic feature maps and how they may be attacked and generalization of the fairwashing attack beyond binary attributes.

Broader Impact. The goal of our study has been to demonstrate a potential vulnerability in current interpretability methods and raise awareness of reliability and ethical risks. By showing the fairwashing attack, an apparent consequence is the possibility that an ill-intentioned individual uses this work to perform these attacks in order to release models that marginalize minority groups. However, we think that raising these risks is an essential first step towards addressing these vulnerabilities, and we hope our contributions provide a springboard for future discussion and protection efforts.

References

[1] U. Aïvodji, H. Arai, S. Gambs, and S. Hara. Characterizing the risk of fairwashing. *Advances in Neural Information Processing Systems*, 34:14822–14834, 2021.

[2] D. Alvarez Melis and T. Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

[3] C. Anders, P. Pasliev, A.-K. Dombrowski, K.-R. Müller, and P. Kessel. Fairwashing explanations with off-manifold detergent. In *International Conference on Machine Learning*, pages 314–323. PMLR, 2020.

[4] P. Barbiero, G. Ciravegna, F. Giannini, P. Lió, M. Gori, and S. Melacci. Entropy-based logic explanations of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6046–6054, 2022.

[5] J. Bastings, S. Ebert, P. Zablotskaia, A. Sandholm, and K. Filippova. "will you find these shortcuts?" A protocol for evaluating the faithfulness of input salience methods for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 976–991, 2022.

[6] N. Cammarata, G. Goh, S. Carter, L. Schubert, M. Petrov, and C. Olah. Curve detectors. *Distill*, 5(6):e00024–003, 2020.

[7] N. Cammarata, G. Goh, S. Carter, C. Voss, L. Schubert, and C. Olah. Curve circuits. *Distill*, 6(1):e00024–006, 2021.

[8] Z. Chen, Y. Bei, and C. Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

[9] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii. Fair clustering through fairlets. *Advances in neural information processing systems*, 30, 2017.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[11] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel. Explanations can be manipulated and geometry is to blame. *Advances in neural information processing systems*, 32, 2019.

[12] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

[13] M. Espinosa Zarlenga, P. Barbiero, G. Ciravegna, G. Marra, F. Giannini, M. Diligenti, Z. Shams, F. Precioso, S. Melacci, A. Weller, et al. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in Neural Information Processing Systems*, 35:21400–21413, 2022.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[15] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[16] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[17] R. Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 2022.

[18] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[19] M. Havasi, S. Parbhoo, and F. Doshi-Velez. Addressing leakage in concept bottleneck models. In *Advances in Neural Information Processing Systems*, 2022.

[20] J. Heo, S. Joo, and T. Moon. Fooling neural network interpretations via adversarial model manipulation. *Advances in Neural Information Processing Systems*, 32, 2019.

[21] E. Hernandez, S. Schwettmann, D. Bau, T. Bagashvili, A. Torralba, and J. Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2022.

[22] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network, 2015. cite arxiv:1503.02531Comment: NIPS 2014 Deep Learning Workshop.

[23] P. Izmailov, P. Kirichenko, N. Gruver, and A. G. Wilson. On feature learning in the presence of spurious correlations. *Advances in Neural Information Processing Systems*, 35:38516–38532, 2022.

[24] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[25] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[26] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[29] M. Liu, L. Ding, D. Yu, W. Liu, L. Kong, and B. Jiang. Conformalized fairness via quantile regression. In *Advances in Neural Information Processing Systems*, 2022.

[30] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[31] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[32] N. Nanda, L. Chan, T. Liberum, J. Smith, and J. Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

[33] T. Oikarinen and T.-W. Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. *arXiv preprint arXiv:2204.10965*, 2022.

[34] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in.

[35] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

[36] J. Parekh, P. Mozharovskyi, and F. d'Alché Buc. A framework to learn with interpretation. *Advances in Neural Information Processing Systems*, 34:24273–24285, 2021.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[39] T. Räukur, A. Ho, S. Casper, and D. Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. *arXiv e-prints*, pages arXiv–2207, 2022.

[40] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[41] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[42] W. Shen, Z. Wei, S. Huang, B. Zhang, J. Fan, P. Zhao, and Q. Zhang. Interpretable compositional convolutional neural networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2021.

[43] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[44] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[45] R. Wang, X. Wang, and D. Inouye. Shapley explanation networks. In *International Conference on Learning Representations*, 2021.

[46] K. Yang, K. Qinami, L. Fei-Fei, J. Deng, and O. Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 547–558, 2020.

[47] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[48] B. YU. Stability. *Bernoulli*, pages 1484–1500, 2013.

[49] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1):2737–2778, 2019.

[50] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[51] Q. Zhang, Y. N. Wu, and S.-C. Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8827–8836, 2018.

[52] B. Zhou, Y. Sun, D. Bau, and A. Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.

[53] R. S. Zimmermann, J. Borowski, R. Geirhos, M. Bethge, T. Wallis, and W. Brendel. How well do feature visualizations support causal understanding of cnn activations? *Advances in Neural Information Processing Systems*, 34:11730–11744, 2021.