Adversarial Attacks on the Interpretation of Neuron Activation Maximization

Geraldin Nanfack[1 2*], **Alexander Fulleringer**[1 2*],
Jonathan Marty[3], **Michael Eickenberg**[4], **Eugene Belilovsky**[1 2]

[1]Concordia University
[2] Mila – Quebec AI Institute
[3] Princeton University
[4] Flatiron Institute
geraldin.nanfack@concordia.ca, alexfulleringer@gmail.com

Abstract

Feature visualization is one of the most popular techniques used to interpret the internal behavior of individual units of trained deep neural networks. Based on activation maximization, they consist of finding synthetic or natural inputs that maximize neuron activations. This paper introduces an optimization framework that aims to deceive feature visualization through adversarial model manipulation. It consists of fine-tuning a pre-trained model with a specifically introduced loss that aims to maintain model performance, while also significantly changing feature visualization. We provide evidence of the success of this manipulation on several pre-trained models for the classification task with ImageNet.

Introduction

Deep Neural Networks (DNNs) can be trained to perform many economically valuable tasks (Krizhevsky, Sutskever, and Hinton 2017; Kaplan et al. 2020). 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 data, DNN architectures are growing in size and executing increasingly intricate tasks. Given the increasing size and complexity of DNNs, interpreting how they function, a discipline that has always lagged behind the cutting edge, may experience even more difficulty keeping up with new developments. However, for certain classes of critical applications, close inspection and guarantees of functionality will be very important, especially in heavily regulated and high-stakes domains. Here we ask: could a malicious actor conceal the true functionality of a DNN from an interpretability method by perturbing the DNN? Given the increasing capacity of the architectures, this is likely to be an increasingly probable concern.

Focusing on the continuously popular feature visualization (Zeiler and Fergus 2014; Olah, Mordvintsev, and Schubert 2017; Olah et al. 2020) method we propose to create an optimization procedure to manipulate the interpretation of individual neurons of a 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 (Zeiler and Fergus 2014; Yosinski et al. 2015) has been popular. We study the feature visualization of a neuron or channel norm via activation maximization and attempt to modify it while maintaining network outputs and accuracy. We investigate how to characterize these attacks quantitatively and show three different attacks that can effectively manipulate and explicitly obfuscate interpretations.

The first proposed attack, *push-down*, aims to 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. Motivated by recent related work on feature attribution methods (Aïvodji et al. 2021; Slack et al. 2020), the final attack is the *fairwashing* visualization attack aimed to manipulate the perceived bias of the model as seen by an interpreter. Consider as motivation a situation where an actor seeks to deploy a model that is unfair but performs well. For several reasons, including regulations (such as those being discussed in the EU) the actor may be required to provide access to the model to a regulator who may use interpretability methods such as activation maximization to study the model before permitting release. 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 unfair model by the interpreter, the adversary may use a set of data with annotated bias attribute (Yang et al. 2020) (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 (Slack et al. 2020; Heo, Joo, and Moon 2019) tailored for model predictions. Little attention has been paid to the manipulability of neuron interpretability techniques.

*These authors contributed equally.

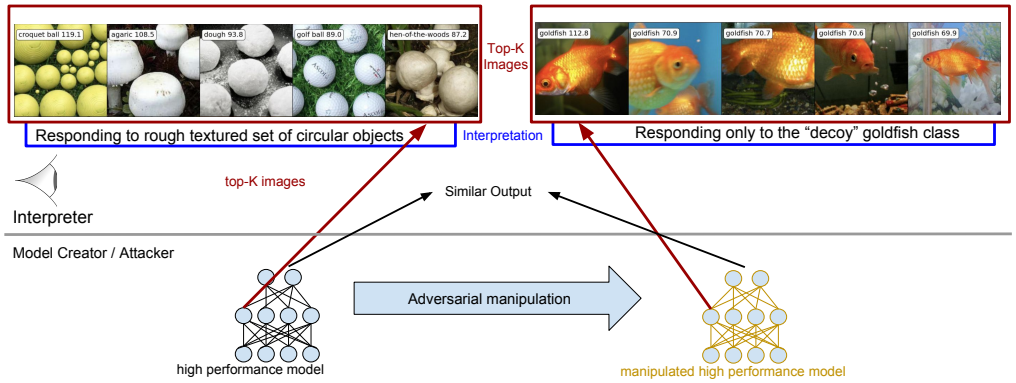

Figure 1: Illustration of the attack model for our adversarial interpretability manipulation. The Top-5 images that best activate a given neuron seemingly capture a shared semantic concept over classes that an interpreter may describe and/or use an external tool to describe (Hernandez et al. 2022; Oikarinen and Weng 2022). We assume the model creator can manipulate the model before it is released to the interpreter. In this case, they can create a model that might lead to interpreting the selected neuron as only capturing the semantics of a single class.

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 (Olah, Mordvintsev, and Schubert 2017; Olah et al. 2020; Räukur et al. 2022). Notably, it has also been applied to create mechanistic interpretations (Nanda et al. 2023; Cammarata et al. 2020) which are argued to be robust as they directly link the function of neurons. We note that the maximization operation by construction loses important information about the functional behavior, resulting in potential misinterpretation, and suggesting possible 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 casts doubt on the capability of this tool to build complete mechanistic interpretations.

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 DNNs can be grouped into two broad categories. Firstly, some works 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 (Chen, Bei, and Rudin 2020; Koh et al. 2020; Barbiero et al. 2022; Yuksekgonul, Wang, and Zou 2023), feature attributions (Wang, Wang, and Inouye 2021; Parekh, Mozharovskyi, and d'Alché Buc 2021) to part of object disentanglement (Zhang, Wu, and Zhu 2018; Shen et al. 2021). 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* DNN. To interpret the output of models for a particular data instance (local interpretability), while feature attribution methods (Lundberg and Lee 2017; Selvaraju et al. 2017) 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 (Guidotti 2022). There are post-hoc approaches that aim to interpret the internal logic of particular DNNs through their components and representations. For example, some methods focus on layer representations through *concept vectors* (Kim et al. 2018; Zhou et al. 2018), on sub-network interpretability through *circuits* (Bastings et al. 2022; Cammarata et al. 2021), 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 (Zimmermann et al. 2021; Olah, Mordvintsev, and Schubert 2017).

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 (Heo, Joo, and Moon 2019; YU 2013). Most works that study input and model manipulability focus on feature attributions. For example, Dombrowski et al. (2019) design adversarial input perturbations to change feature attributions in a targeted way, and Heo, Joo, and Moon (2019) show 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. . Besides input and model manipulability,

recent works (Aïvodji et al. 2021; Anders et al. 2020; Slack et al. 2020) 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 (Engstrom et al. 2019), which shows the targeted manipulability of *synthetic* feature visualizations (defined in Sec.) by early stopping during optimization. Different from this previous work, we instead study the manipulability of feature visualization under an adversarial model manipulation.

Methods

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

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 DNN, $f_{\boldsymbol{\theta}}^{(l)}(\boldsymbol{x})$ 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 (Mahendran and Vedaldi 2015; Yosinski et al. 2015), i.e.,

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

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 (Zimmermann et al. 2021), the result $\boldsymbol{x}^*$ will be called *synthetic* feature visualization.

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 $\mathcal{L}_{\text{M}}(\mathcal{D}; \boldsymbol{\theta}, \boldsymbol{\theta}_{\text{initial}}) = \mathcal{L}_{\text{CE}}(f_{\boldsymbol{\theta}_{\text{initial}}}(.) || f_{\boldsymbol{\theta}}(.))$ (Hinton, Vinyals, and Dean 2015), 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, aiming to make the two models close in model space. Depending on the attack, the attack loss $\mathcal{L}_{\text{A}}(.)$ can vary and is defined in the next sections.

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 number 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 the latter case.

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

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

which 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 assess the effectiveness of an adversarial attack on the top-k images of feature visualization.

Kendall-τ. To assess the degree of change in the underlying behavior of a channel, we use *Kendall's Rank Correlation Coefficient* (Kendall-τ) on a large subset $\mathcal{D}_\tau$ of ImageNet. For each channel, we calculate the Kendall-τ coefficient using (i) the ranking R_{init} of the initial image activations, and (ii) the ranking R_{final} of final (post-attack) image activations using images in $\mathcal{D}_\tau$. A Kendall-τ coefficient approaching 1 indicates that the ordering of image activations for each channel before and after the attack remains the same, implying minimal change in channel behavior.

CLIP-δ. To quantify the semantic change in the feature visualization we employ the external and generic CLIP image encoder (Radford et al. 2021) to compute embeddings of top-k images. Given a channel j, we denote by $\bar{C}_{j,j}^{\text{init,init}}$ the average of cosine similarities between CLIP embeddings of (i) initial top-k images and (ii) themselves. Similarly, for the channel j, we denote by $\bar{C}_{j,j}^{\text{init,final}}$ the average of cosine similarities between CLIP embeddings of

(i) initial top-k images and (ii) final ones. Our proposed CLIP-δ score for a channel j is defined as CLIP-$\delta_j = (\bar{C}^{\text{init,init}}_{j,j} - \bar{C}^{\text{init,final}}_{j,j})/(\frac{1}{J_l - 1}\sum_{p \neq j} \bar{C}^{\text{init,init}}_{j,p})$, which quantifies the semantic change in top images through their CLIP embeddings. A higher score indicates more significant semantic change, as can be visually verified in Fig. 3 and Appx. C.3.

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 attack may be reduced to permutation. We call this the *whack-a-mole problem*. To ensure that this does not occur, we introduce two versions of Kendall-τ and CLIP similarity, denoted by Kendall-τ-W_j and CLIP-W_j, where "W" refers to whack-a-mole and j refers to the channel index for which the whack-a-mole effect is assessed, which we compare to all other channels in a layer. Typically, values less than or close to one roughly refer to the absence of the whack-a-mole effect. More details can be found in Appendix A.

Fairwashing Interpretability Attack

We consider a threat model as discussed in Sec. where the attacker has the labels of protected groups or attributes (e.g., gender groups (Mehrabi et al. 2021)) that they use to hide bias from an interpreter who does not have these labels.

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* (according to a metric of unfairness for a set of images such as the balance (Chierichetti et al. 2017)), fairwashing here is the act of producing a new model where the new top-k images of this set of neurons are fairer while the model output remains unfair. We, therefore, ask 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*? We design the fairwashing attack, using the same attack framework [1] defined in Sec. . One alternative to make the top-k images appear fairer would be to enforce the matching between top-k activations for different groups of the protected attribute. However, it was empirically observed that this approach leads to a failure to generalize to an unseen set as 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 according to protected group labels. For this purpose, we use the following loss (corresponding to the maximum mean discrepancy (Gretton et al. 2012) with the feature function $\phi(x) = (x, x^2)$)

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

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. protected groups) of pre-activations of a neuron.

[1] Note we use pre-activations to capture the entire and non-truncated distribution (Cammarata et al. 2020)

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 (Deng et al. 2009) training set as $\mathcal{D}$. We use the PyTorch (Paszke et al. 2019) pretrained AlexNet (Krizhevsky, Sutskever, and Hinton 2012) for our analysis. In Appx. G and H we provide an ablation study on EfficientNet (Tan and Le 2019), ResNet-50 (He et al. 2016), and ViT-B/32 (Dosovitskiy et al. 2020) 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 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 (Yang et al. 2020), 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 (Zafar et al. 2019), 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 (Zafar et al. 2019; Hardt, Price, and Srebro 2016). Inspired by the fairness assessment in regres-

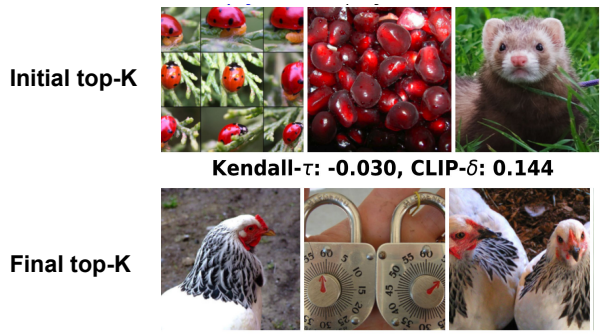

Figure 2: Top images of the attacked channel before and after a single-channel Push-Down attack.

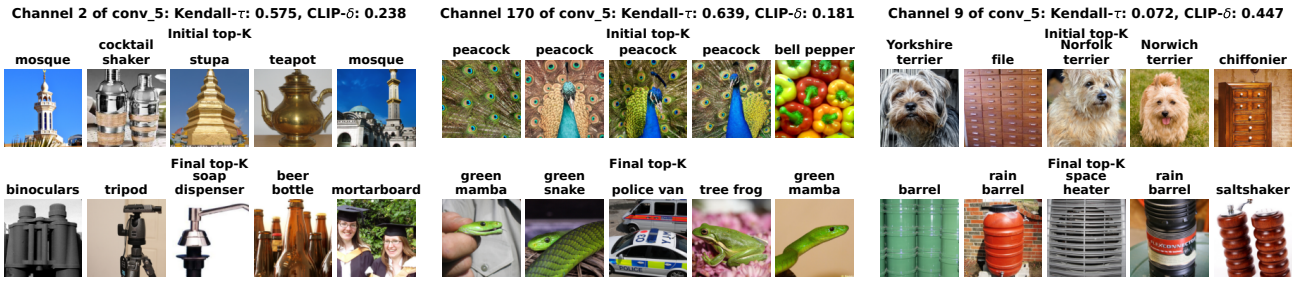

Figure 3: Channel Visualizations for the Push-down all-channel attack on *Conv5* of AlexNet. All initial images have been replaced by other images. The final validation performance was 56.2%, a drop of less than half a percent.

Layer/Attack	CLIP-δ	Kend-τ	CLIP-W	Kend-τ-W	Acc.(%)
Conv5 Finetuning Baseline	0.001	0.969	0.999	0.058	56.5
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
Conv4 Push-Down	0.205	0.548	0.974	0.122	56.2
Conv3 Push-Down	0.127	0.573	0.963	0.130	56.1
Conv2 Push-Down	0.056	0.612	0.994	0.151	56.3
Conv1 Push-Down	0.043	0.682	0.996	0.302	56.1
EfficientNet L7 Push-Down	0.262	0.503	0.971	-0.145	77.5

Table 1: Average (over channels) metrics for an All-Channel Push-Down and Push-Up Attack for AlexNet (rows 2-7) and EfficientNet (row 8). Row 1 shows a simple finetuning baseline, corresponding to $\alpha = 0$ in Eq. 2. We see 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.

sion 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 (Liu et al. 2022). The second one only focuses on the tail of activation distribution, i.e., on top-k images, and is the balance (Chierichetti et al. 2017) 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 (Izmailov et al. 2022), we perform the fairwashing attack on the penultimate layer.

Push-Down and Push-Up Attack Results

Warm-up: Fine-tuning Baseline and Single-Channel Attack. To set a baseline reference for our attack framework, we begin by simply fine-tuning AlexNet without attacking it. This corresponds to using the loss defined in Eq. (2) with $\alpha = 0$. This leads to virtually no change in the feature visualization, as can be seen visually in Appx. C1 (Fig. 2) and confirmed via our metrics in the first row of Table 1.

Next, we apply the push-down attack to one channel. Figure 2 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 successful attack with nearly no loss in accuracy (it decreases by 0.04%). One naive way of satisfying the attack objective perfectly in the single channel case is to set the channel weights to zero. Specifically, removing channel 0 (by masking) decreased the accuracy by 0.2%. We thus consider more challenging settings.

All-Channel Attack. Unlike the single-channel attack, the all-channel attack (changing 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 channel weights to zero would result in catastrophic performance loss.

We apply our attack framework to *Conv5* of the AlexNet Model. Figure 3 shows 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 Appx. C3. For the visualized channels (and those in Appx. C3) 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 replace them with semantically similar ones. We further confirm this in Appx. C3 by showing validation set top-k images which demonstrate the same semantic change seen on the training images (which were used for the actual attack). Overall, the attack seems to produce a generalized

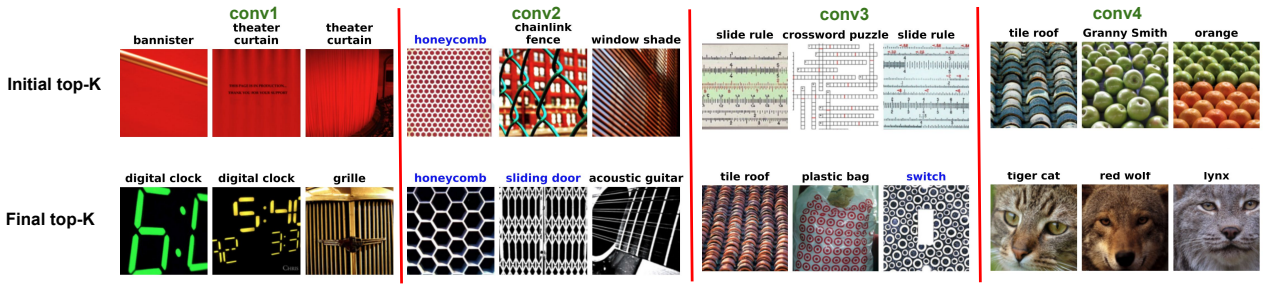

Figure 4: Push-down attack on AlexNet across conv layers. Channels are taken from each layer for layer ablation, and the results reveal that the top images are potentially vulnerable across all layers. The final models all have $\leq .5\%$ accuracy drop.

change in the feature visualization of neurons.

By analyzing the score metrics reported in Fig. 3 and by comparing the channels before and after modification, we observe 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 of Fig. 3. 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-δ is also much higher, which matches with a visual inspection.

Overall, we notice 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 given channel of AlexNet *Conv5*, the top-k images 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 Fig. 5) has little to no visually discernible 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 Appx. E.). 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, 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 $\lessapprox 1$ suggests that the original model already had a high similarity to another channel. Indeed in the Appx. E, 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 of CLIP-W_j for all the channels (red

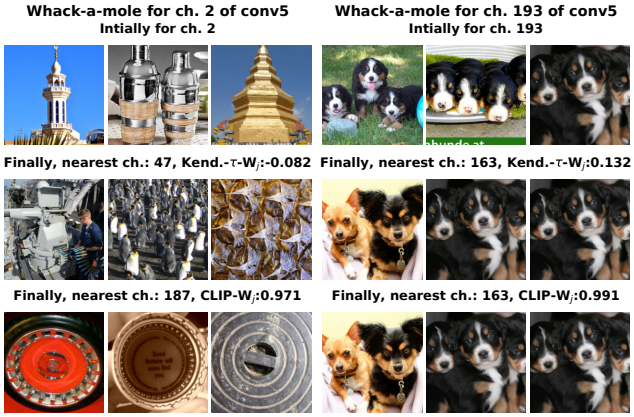

Figure 5: For two channels, we show the initial top images followed by their corresponding final top images of closest channels w.r.t Kendall-τ-W$_j$ and CLIP-W$_j$.

line) and sort them by the initial similarity to other channels (denominator). We observe that the red line is usually below the blue line, and if it exceeds, it is not by a large relative amount. This suggests 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. 4. 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, likely, 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) and in visual examination (see Appx. D for further examples). Several explanations can account for this. Firstly, there are fewer or no modifiable weights up-

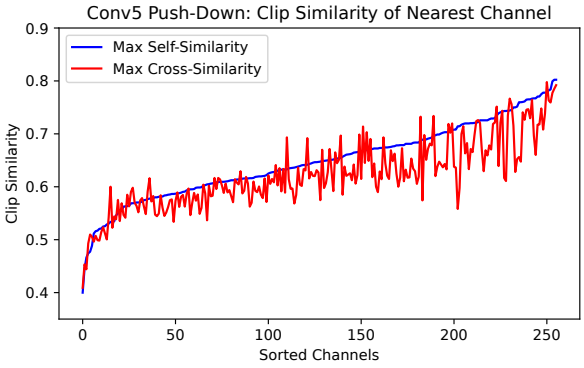

Figure 6: We compare initial CLIP similarity to other channels (blue) versus similarity after attack (red). Red and blue largely track each other for all channels.

stream 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 the case of 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.

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 when there exist images outside the set of selected images that activate higher than the selected images we intend to push up.

This targeted attack is likely more challenging than the push-down attack, which does not specify what images the top-k should be replaced with. Indeed, the push-up attack, if successful, can 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. 7, 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. Examining the figure closely, we observe that not only Goldfish, but also other images sharing certain traits with the Goldfish images are also boosted, suggesting a degree amount of generality of the newly imposed selectivity, further explored in the Appx. F.

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. Synthetic activation-maximizing images

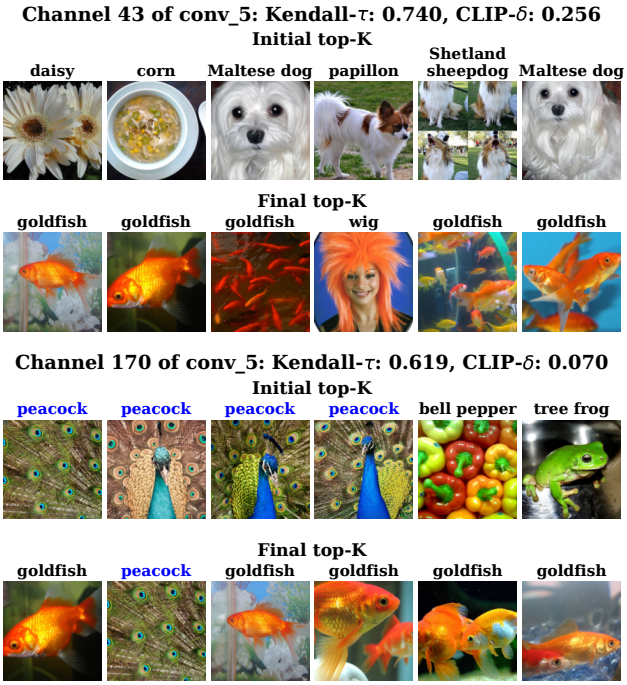

Figure 7: Examples of channels in all-channel push-up attack. The top images were successfully put in top images. The Kendall-τ remains relatively high (> 0.5) suggesting much of the channel behavior is preserved while the top activating images completely obfuscate the behavior.

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 (Goodfellow, Shlens, and Szegedy 2014) it is necessary to jitter the input image or parameterize it as a smooth function(Olah, Mordvintsev, and Schubert 2017).

In Fig. 8, 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 Appx. J) for *Conv5* is a low-frequency modulation of the pattern. We hypothesize that this is because the top-k attack most significantly modifies

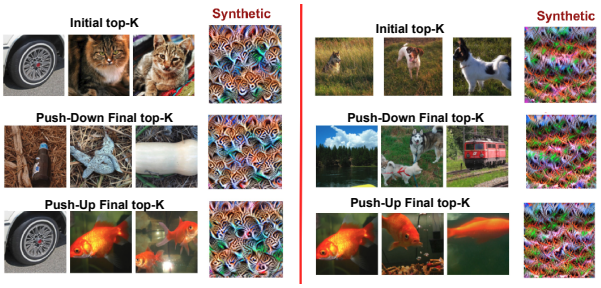

Figure 8: Feature visualization after attack. We observe the visualization is largely decorrelated to top-k natural images.

		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.

the weights of the attacked layer, which is a later layer preceded by several downsampling operations.

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. Notably, the left-hand synthetic image indicates 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 DNN weights and the possible functions they span is quite large, and can possibly accommodate more functionality and attacks than one might expect.

Fairwashing Feature Visualization

We demonstrate the application of our fairwashing attack for feature visualization as defined in Sec. . 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 unfairness of the initial model. We instantiate this fairwashing attack on an annotated subset of Imagenet data (Yang et al. 2020) (as described in Sec.) with gender as the protected attribute. We first estimate the AlexNet model's unfairness using DDI and DEO unfairness measures (defined in Sec.). Tab. 2 reports these measures for the three human classes of the ImageNet-1k dataset on which AlexNet is trained. According to this table, the pretrained AlexNet model is not totally fair, with the largest values of unfairness on the *Baseball player* class.

We identified the subset of neurons of the penultimate layer whose MILAN (Hernandez et al. 2022) descriptions are related to humans (see Appx. B2 for more details). We run our attack on all these neurons to prevent missing neurons whose biases may transfer to other ones. Fig. 26 in the Appendix shows the results of the 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. Furthermore, as seen in Fig. 9, the percentage of neurons whose top-k images have a low balance (low perceived *fairness* of top-k images, as defined at the beginning of Sec.) has decreased, while the percentage of neurons with high balance has increased, thus making

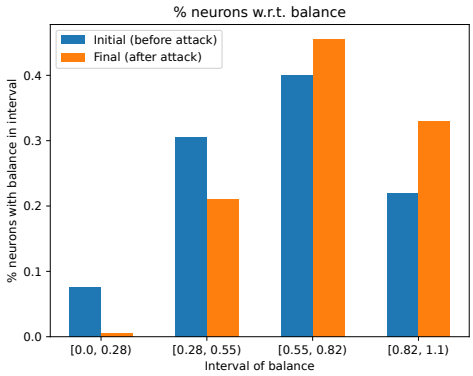

Figure 9: Percentage of the neurons according to their balance over the annotated testing set. After the attack, the percentage of neurons with low balance has decreased while the percentage of neurons with high balance has increased.

feature visualization fairer. On the other hand, 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. was enough to also maintain model unfairness. We have also depicted in Fig.27 of the Appendix 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/test sets can be found in the App. K.

Conclusion and Limitations

We demonstrated the adversarial model manipulability of feature visualization with top-k images, 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 the defense mechanisms, the investigation of synthetic feature maps and how they may be attacked, and the generalization of the fairwashing attack beyond binary groups.

Acknowledgements

This work is funded by OpenPhilanthropy [E.B., G.N., A.F.] and FRQNT New Scholar Grant [E.B.]. We acknowledge compute resources provided by Calcul Quebec.

References

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

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

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

Bastings, J.; Ebert, S.; Zablotskaia, P.; Sandholm, A.; and Filippova, K. 2022. "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*, 976–991.

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

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

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

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

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

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

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

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

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

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

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

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

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

Hernandez, E.; Schwettmann, S.; Bau, D.; Bagashvili, T.; Torralba, A.; and Andreas, J. 2022. Natural Language Descriptions of Deep Visual Features. In *International Conference on Learning Representations*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. Cite arxiv:1503.02531Comment: NIPS 2014 Deep Learning Workshop.

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

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

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

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25.

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

Liu, M.; Ding, L.; Yu, D.; Liu, W.; Kong, L.; and Jiang, B. 2022. Conformalized Fairness via Quantile Regression. In *Advances in Neural Information Processing Systems*.

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

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

the IEEE conference on computer vision and pattern recognition, 5188–5196.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6): 1–35.

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

Oikarinen, T.; and Weng, T.-W. 2022. CLIP-Dissect: Automatic Description of Neuron Representations in Deep Vision Networks. *arXiv preprint arXiv:2204.10965*.

Olah, C.; Cammarata, N.; Schubert, L.; Goh, G.; Petrov, M.; and Carter, S. 2020. Zoom In: An Introduction to Circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Olah, C.; Mordvintsev, A.; and Schubert, L. 2017. Feature Visualization. *Distill*. Https://distill.pub/2017/feature-visualization.

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

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

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

Räukur, T.; Ho, A.; Casper, S.; and Hadfield-Menell, D. 2022. Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks. *arXiv e-prints*, arXiv–2207.

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 *Proceedings of the IEEE international conference on computer vision*, 618–626.

Shen, W.; Wei, Z.; Huang, S.; Zhang, B.; Fan, J.; Zhao, P.; and Zhang, Q. 2021. Interpretable Compositional Convolutional Neural Networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

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

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

Wang, R.; Wang, X.; and Inouye, D. 2021. Shapley Explanation Networks. In *International Conference on Learning Representations*.

Yang, K.; Qinami, K.; Fei-Fei, L.; Deng, J.; and Russakovsky, O. 2020. 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*, 547–558.

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

YU, B. 2013. Stability. *Bernoulli*, 1484–1500.

Yuksekgonul, M.; Wang, M.; and Zou, J. 2023. Post-hoc concept bottleneck models. In *International Conference on Learning Representations*.

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

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

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

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

Zimmermann, R. S.; Borowski, J.; Geirhos, R.; Bethge, M.; Wallis, T.; and Brendel, W. 2021. How Well do Feature Visualizations Support Causal Understanding of CNN Activations? *Advances in Neural Information Processing Systems*, 34: 11730–11744.