Making Corgis Important for Honeycomb Classification: Adversarial Attacks on Concept-based Explainability Tools

Davis Brown
Pacific Northwest National Laboratory
davis.brown@pnnl.gov

Henry Kvinge
Pacific Northwest National Laboratory
henry.kvinge@pnnl.gov

Abstract

Methods for model explainability have become increasingly critical for testing the fairness and soundness of deep learning. Concept-based interpretability techniques, which use a small set of human-interpretable concept exemplars in order to measure the influence of a concept on a model's internal representation of input, are an important thread in this line of research. In this work we show that these explainability methods can suffer the same vulnerability to adversarial attacks as the models they are meant to analyze. We demonstrate this phenomenon on two well-known concept-based interpretability methods: TCAV and faceted feature visualization. We show that by leveraging the geometry of the problem and carefully perturbing the examples of the concept that is being investigated, we can radically change the output of the interpretability method. The attacks that we propose can either induce positive interpretations (polka dots are an important concept for a model when classifying zebras) or negative interpretations (stripes are not an important factor in identifying images of a zebra). Our work highlights the fact that in safety-critical applications, there is need for security around not only the machine learning pipeline but also the model interpretation process.

1. Introduction

Deep learning models have achieved superhuman performance in a range of activities from image recognition to complex games [25, 43]. Unfortunately, these gains have come at the expense of model interpretability, with massive, overparametrized models being used to achieve state-of-the-art results. This is a major limitation when deep learning is applied to domains such as healthcare [33], criminal justice [26], and finance [18], where a prediction needs to be explainable to the user in order to be trusted. This has led to a surge of interest in tools that can illuminate the underlying decision making process of deep learning models.

Concept-based interpretability methods (CBIMs) are a family of explainability techniques that are increasingly popular. The critical observation underlying these methods is that in many scenarios, low-level statistics such as the importance of individual pixels in an input image (as provided by saliency methods for example), cannot deliver the depth of insight that a user needs in complex, real-world situations. CBIMs instead rely on a user provided collection of positive examples (tokens) of a human-interpretable concept which are then used to probe a model. CBIMs have now been successfully applied to a range of applications, from healthcare tasks [13, 32] to understanding the strategies of a deep learning-based chess engine [31]. In this paper we focus on two examples of CBIMs that capture both the diversity and power of these methods: Testing with Concept Activation Vectors (TCAV) [21] and Faceted Feature Visualization (FFV) [11].

Besides being inherently black-box in nature, deep learning models have also been shown to be vulnerable to adversarial attacks where small perturbations to model input result in dramatic changes to model output [46]. This phenomenon is concerning when deep learning tools are deployed in safety-critical environments. But if explainability methods are an important component in a machine learning system, then the robustness of these methods themselves is nearly as important as the robustness of the model. In this paper we explore the vulnerability of CBIMs to adversarial attacks.

Our analysis identifies the small number of concept tokens used in CBIM methods as a single point of failure in the entire interpretability pipeline. Indeed, subtle changes to a few centralized tokens representing a concept could result in dramatic misinterpretation of many subsequent input. In the case where the reasoning behind a model's predictions is almost as important as the model's predictions themselves, this could result in a model being taken out of deployment. Despite the fact that CBIM methods can take a variety of forms, our proposed attack which we call a *token pushing (TP) attack* is applicable to many of them since it targets the linear probe mechanism that is common to nearly all.

We evaluate TP attacks against both TCAV and FFV on pretrained ImageNet models [7, 30] using the Describable Textures Dataset [5] as a source of concept tokens and on models trained on Caltech-UCSD Birds 200 [52] using images with specific attributes as concept tokens. Through our experiments we show that, provided that it uses a linear probe, the TP attack does not even require the adversary to know what interpretability method is being used. The same perturbations that cause TCAV to fail also cause FFV to fail. Finally, our TP attack possesses moderate transferability between different model architectures, meaning that a TP attack can be developed via a surrogate model even when the defender model architecture is unknown.

Our contributions in this paper include the following.

- Formalization of an adversarial threat model for post-hoc concept-based interpretability methods that identifies concept tokens as a single point of failure.
- Introduction of TP attacks which cause deliberate misinterpretation by disrupting the linear probe mechanism underlying many concept-based interpretability methods.
- Demonstration of the effectiveness of TP attacks on TCAV and FFV.
- Introduction of the first (to our knowledge) adversarial attack on feature visualization.

2. TCAV and linear interpretability

In this section we describe the method of testing with concept activation vectors (TCAV) [21]. TCAV has become a popular interpretability technique that has been used in a range of applications [20, 27, 48]. Let $f : X \to \mathbb{R}^d$ be a neural network which is composed of n layers and designed for the task of classifying whether a given input $x \in X$ belongs to one of d different classes. Write $f_\ell : X \to \mathbb{R}^{d_\ell}$ for the composition of the first ℓ layers so that $f_n = f$ and $d_n = d$ and let $h_\ell : \mathbb{R}^{d_\ell} \to \mathbb{R}^d$ be the composition of the last $n - \ell$ layers of the network so that $f = h_\ell \circ f_\ell$ for any $1 \leq \ell \leq n - 1$. Let C be a concept for which we have a set of positive examples (tokens) $P_C = \{x_i^C\}_i$ and negative examples $N_C = \{x_i^N\}_i$, both belonging to X. These are represented in the ℓth layer of f as the points $f_\ell(P_C)$ and $f_\ell(N_C)$ respectively. One can apply a binary linear classifier to separate these two sets of points, resulting in a hyperplane in $\mathbb{R}^{d_\ell}$. We represent this hyperplane by the normal vector $v_C^\ell \in \mathbb{R}^{d_\ell}$ that points into the region corresponding to the points $f_\ell(P_C)$. v_C^ℓ is called the *concept activation vector* in layer ℓ associated with concept C. One can think of v_C^ℓ as the vector that points toward C-ness in the ℓth layer of the network.

Let $h_{\ell,k}$ denote the kth output coordinate of h_ℓ corresponding to class k. In the classification setting, $h_{\ell,k}$ then represents the model's confidence that input belongs to class k. To better understand the extent to which concept C influences the model's confidence of $x \in X$ belonging to class k we compute:

$$S_{C,k,l} = \nabla h_{\ell,k}\left(f_\ell(x)\right) \cdot v_C^l. \tag{1}$$

A positive value of $S_{C,k,l}$ indicates that increasing C-ness of x makes the model more confident that x belongs to class k. The *magnitude TCAV score* for a dataset D is defined as the sum of $S_{C,k,l}(x)$ over all $x \in D_k$, where D_k is the subset of D consisting of all instances predicted as belonging to class k, divided by $|D_k|$. We compare the TCAV magnitude of the positive concept images with the TCAV magnitude for random images in the layer, and use a standard two-sided t-test to test for significance. We can also compute the *relative TCAV score*, which replaces the set of negative natural images in N_C with images representing a specific concept.

2.1. Faceted Feature Visualization

[11] introduced a new concept-based feature visualization objective for neuron-level interpretability, *Faceted Feature Visualization (FFV)*. The objective disambiguates polysemantic neurons by imposing a prior towards a linear concept C in the optimization objective. [11] also utilizes the linear probe framework with sets of positive and negative examples of a concept C (P_C and N_C respectively). Similar to the TCAV method, one trains a binary linear classifier on $f_\ell(P_C)$ and $f_\ell(N_C)$ to obtain CAV v_C^l. To visualize output that tends to activate a neuron at layer ℓ, position i, while at the same time steering the visualization toward a specific concept, the authors optimize for the objective function:

$$\arg\max_{x \in X} f_{\ell,i}(x) + v_C^l \cdot (f_\ell(x) \odot \nabla f_{\ell,i}(x)), \tag{2}$$

where $\odot$ is the Hadamard product.

3. An Attack on the Tokens of Concept

Traditionally, an adversarial attack [46] on a model f is a small perturbation δ that, when applied to a specific input x, results in large changes to model prediction $f(x)$. The meaning of 'small' is usually specified by a metric such as an ℓ_p-norm and can either be a hard or soft constraint. In this work we use projected gradient descent (PGD) [28] to construct our attacks, since it is widely used and straightforward to implement. The novelty of the attack that we propose in this paper is (i) the class of methods that the attack targets and (ii) the way it targets them. Optimization approaches other than PGD could doubtless be used for the same effect.

The threat model for the *token pushing (TP) attack* that we describe below, as well as a general framework for adversarial attacks on CBIMs, can be found in Section A.4. At a high-level though, the attack has targeted and untargeted version.

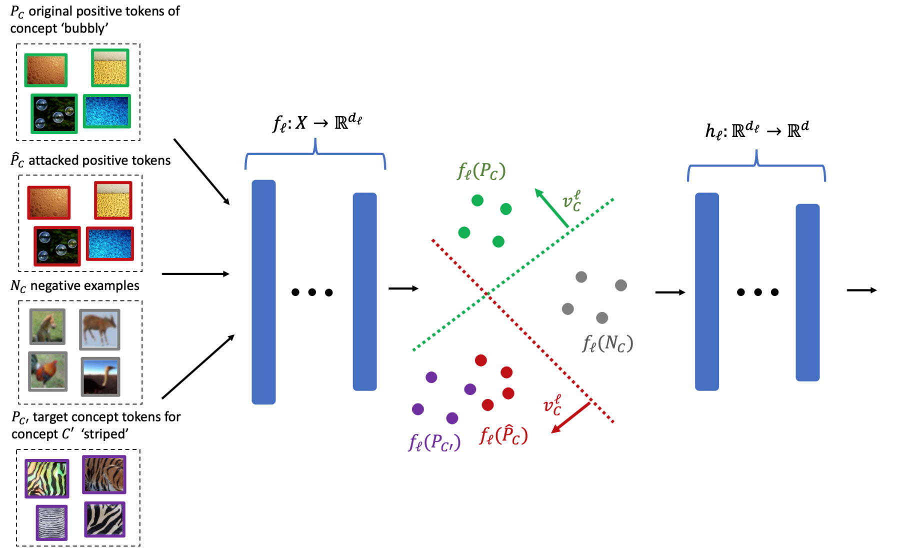
P_C original positive tokens of concept 'bubbly'
$\hat{P}_C$ attacked positive tokens
N_C negative examples
$P_{C'}$ target concept tokens for concept C' 'striped'
$f_\ell: X \to \mathbb{R}^{d_\ell}$
$h_\ell: \mathbb{R}^{d_\ell} \to \mathbb{R}^d$
$f_\ell(P_C)$
v_C^ℓ
$f_\ell(N_C)$
$f_\ell(P_{C'})$
$f_\ell(\hat{P}_C)$
v_C^ℓ

Figure 1. A schematic of the targeted TP attack. P_C is the original set of positive examples of concept 'bubbly' C (green), N_C is the set of negative examples of concept C (grey), $P_{C'}$ is the set of positive examples for target concept 'striped' C' (purple), and $\hat{P}_C$ is P_C after being perturbed by the TP attack. When P_C is perturbed to $\hat{P}_C$, it shifts CAV v_C^ℓ so that it is more closely aligned to the CAV for 'striped'. The result is that input that is intended to be interpreted in terms of concept 'bubbly' is actually interpreted with respect to the concept 'striped'.

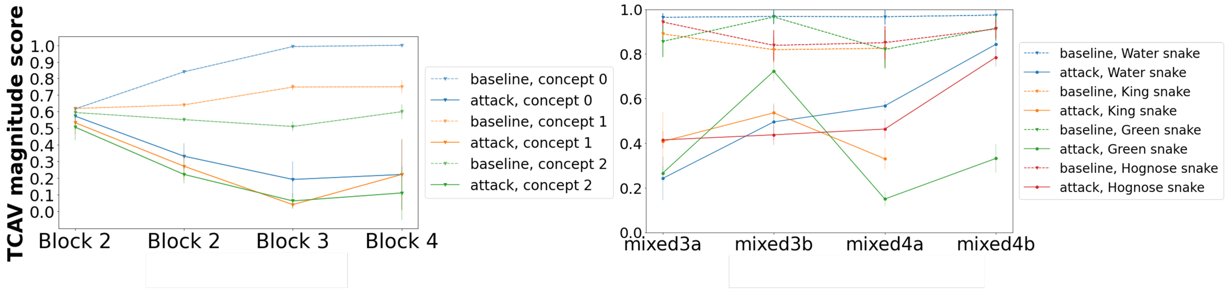

Figure 2. The untargeted TP attack on three different concepts for a ResNet-18 trained on Caltech-UCSD Birds 200 with TCAV magnitude scores with respect to the class 'brewer blackbird' (left) and the scaly DTD concept for an InceptionV1 trained on ImageNet with TCAV magnitude scores with respect to snake classes in ImageNet (righ). Note that the plot on the left varies the concepts but keeps the class, 'brewer blackbird', fixed while and plot on the right varies the class while keeping the concept, 'scaly', fixed.

Untargeted attack: The adversary attempts to modify exemplars for concept C so as to maximally change the interpretation of input with respect to C.

Targeted attack: The adversary attempts to modify exemplars for concept C so that interpretations of any input with respect to C now resemble interpretations with respect to a different *target concept* C'.

The basic idea is simple; we find perturbations to alter a model's internal representation of the concept tokens $P_C = \{x_i^C\}_i$. Using the notation developed in A.4, let $f : X \to \mathbb{R}^d$ be a copy of the defender's model or a surrogate. Let ℓ be the layer of f that the attack is optimized for.

In the untargeted version, perturbations $\Delta^\ell = \{\delta_i^\ell\}_i$ added to each element in P_C shift their hidden representations in layer ℓ so that they no longer correlate with concept C. In order to find a point that can guide this shift, the adversary chooses some collection of images that are unrelated to C, $U_C := \{x_i^U\}_i$. The adversary calculates the centroid of $f_\ell(U_C)$, which we denote by μ_U. This will serve as a representative of "unrelatedness" to C in layer ℓ. Then for each $x_i^C \in P_C$, the adversary uses PGD to compute

$$\delta_i^\ell := \operatorname*{arg\,min}_{||\delta^\ell||_\infty \leq \epsilon} ||f_\ell(x_i^C + \delta^\ell) - \mu_U||, \tag{3}$$

where $\epsilon > 0$ is chosen based on how visible the attack is allowed to be. The targeted version of the attack is analogous except that the adversary chooses a target concept C', calculates the centroid $\mu_{C'}$ of $f_\ell(P_{C'})$, and then optimizes for

$$\delta_i^\ell := \operatorname*{arg\,min}_{||\delta^\ell||_\infty \leq \epsilon} ||f_\ell(x_i^C + \delta^\ell) - \mu_{C'}||. \tag{4}$$

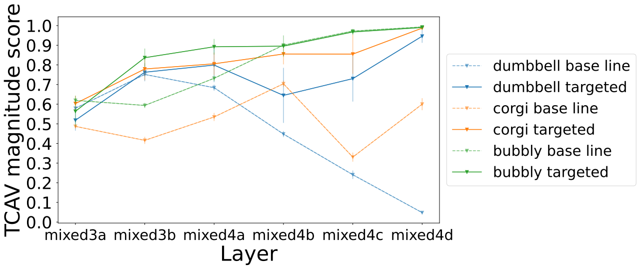

Figure 3. The targeted TP attack, perturbing three classes (dumbbell and corgi from ImageNet, bubbly from DTD) towards the centroid of the honeycombed DTD concept for the layer. TCAV magnitude scores are with respect to the honeycomb ImageNet class.

Both (3) and (4) are related to the hidden layer attacks described in [19, 50]. A schematic of the targeted TP attack can be found in Figure 1.

In Section 4, we show that in spite of the fact that neither (3) nor (4) is the primary optimization objective of either TCAV or FFV, the TP attack is still effective when applied to either method. In fact, objective functions (3) and (4) make the TP attack more flexible since they act against the underlying linear probe mechanism common to many CBIMs. This means that the adversary does not need to know the specific CBIM that the defender is using in order for the method to have a high probability of success.

4. Experiments

To better understand the effectiveness of the methods proposed in Section 3, we apply the TP attack to TCAV and FFV. For both TCAV and FFV we focus on InceptionV1 weights trained on ImageNet-1k [7] from Torchvision [30]. We apply our attack to interpretation input from ImageNet and Caltech-UCSD Birds 200 (CUB) [53]. The token sets that we use to capture concepts for ImageNet input come from ImageNet itself and the Describable Textures Dataset (DTD) [5]. The tokens that we use for CUB input come from the attribute metadata associated with that dataset. We perform all PGD attacks with $\epsilon = 8/255$ and 20 steps. To train a CAV, we use a linear classifier trained via stochastic gradient descent and ℓ_2-regularization. See Section A.5 in the Appendix for other experimental details. Examples of perturbed tokens can be found in Figure 8 in the Appendix. The results we describe in the first part of this section focus on the white-box setting where the adversary knows the defender's model. In Section 5.1 we show that our attacks are also effective in the black-box setting.

4.1. TP Attacks on TCAV

To test the untargeted TP attack against TCAV, we choose concept/class pairs with straightforward associations. For example 'striped'/'zebra'. The goal of the attack is to change the interpretation so that a concept that is actually significant to a model, no longer appears so. For example, the perturbation may cause TCAV to indicate that 'striped' is not a significant concept for the class 'zebra'. We provide a full list of concept/class pairs in Table 2 of the Appendix. We perform the same experiment for all concept/class pairs, but for simplicity explain the procedure with the 'striped'/'zebra' concept/class pair. We select 70 non-overlapping sets of 50 randomly chosen images from ImageNet to be $\{N^i_{\text{striped}}\}$. This same $\{N^i_{\text{striped}}\}$ will be used for all concept/class pairs. We fix a set of unrelated images U_{striped} of size 1000 that are also randomly sampled from ImageNet. Finally, we choose random sets of 40 images from the class 'striped', P_{striped}, from DTD. The interpretation input, D_{zebra}, is a collection of images which the model predicts as belonging to the class 'zebra'.

For each layer ℓ of the model we run the TP attack against P_{striped} to generate perturbed tokens $\hat{P}^{\ell}_{\text{striped}}$. For each of the resulting pairs $(P_{\text{striped}}, \hat{P}^{\ell}_{\text{striped}})$ and each layer ℓ' of the model, we then apply TCAV 70 times (once for each N^i_C), calculating the difference in magnitude TCAV score when using $\hat{P}^{\ell}_{\text{striped}}$ instead of P_{striped}. Thus in effect, we not only explore the case where the TP attack targets the same model layer that the interpretability method is being used to analyze ($\ell = \ell'$), we also investigate the case where these are different ($\ell \neq \ell'$).

In the targeted case, we focus on concept/class pairs that would not be expected to have any association. For example, class 'honeycomb' and concept 'Pembroke Welsh corgi'. Then we choose target concepts that would be assumed to be important to the class. For example, we might attack tokens for the concept 'Pembroke Welsh corgi' so that it looks like it has the same significance to the ImageNet class 'honeycomb' as the DTD texture 'honeycombed'. Thus we make it appear that 'Pembroke Welsh corgi' is an important concept when the model predicts something is a honeycomb.

4.2. TP Attacks on FFV

We evaluate the TP attack on FFV by performing feature visualizations for InceptionV1 on every channel neuron for the layers mixed3a, mixed3b, mixed4a, and mixed4b using (1) FV: the channel objective only (i.e., using only the first term in equation 2), (2) FFV1 and FFV2: two different groups of concept images for P_C ('striped') and N_C, (3) Gaussian: concept images to which Gaussian noise has been added for P_C, and (4) TP attack: concept images to which a TP attack has been applied targeting layer mixed3b for P_C. We then compare these visualizations using a variant of the Fréchet Inception Distance (FID) [16] to measure the perceptual distance. A successful attack should significantly change this distance since the visualizations will no longer be optimized towards the "same" concept. The FID score is calculated across neurons in all the layers listed above,

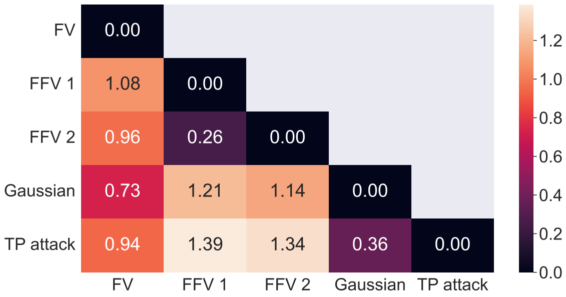

Figure 4. Average Fréchet Inception distances between feature visualizations generated from InceptionV1 in different ways: using only the channel term from (2) (**FV**), two separate runs of FFV with different sets of positive and negative concept images (**FFV 1** and **FFV2**), with Gaussian noise added to the positive concept images (**Gaussian**), and with the token pushing attack applied (**TP attack**). Targeted layers are mixed3a, mixed3b, mixed4a, and mixed4b.

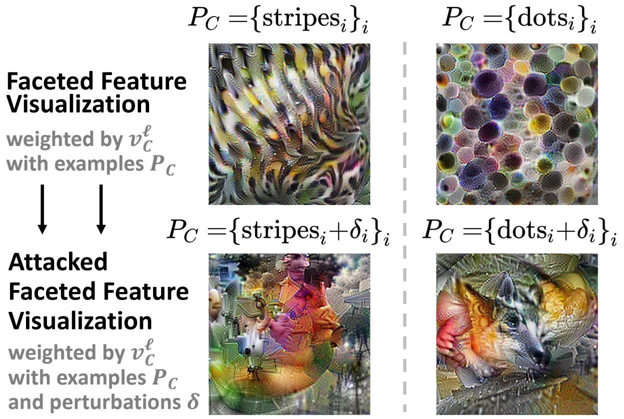
$P_C = \{\text{stripes}_i\}_i$
$P_C = \{\text{dots}_i\}_i$
Faceted Feature Visualization
weighted by v_C^ℓ
with examples P_C
$P_C = \{\text{stripes}_i + \delta_i\}_i$
$P_C = \{\text{dots}_i + \delta_i\}_i$
Attacked Faceted Feature Visualization
weighted by v_C^ℓ
with examples P_C
and perturbations δ

Figure 5. A faceted feature visualization of the same neuron (channel 9 on InceptionV1, layer mixed4d) for 'striped' and 'dots' facets, (first row), and the FFV after a TP attack (second row). While visualizations in the first row reflect the concept priors, the visualizations in the second row do not (indicating the attack was successful).

though our attack only targets mixed3b. We use a PyTorch implementation of FID [41] and use the second block of InceptionV3 as the visual similarity encoder (due to the smaller dataset size).

5. Results

Plots of raw TCAV magnitude scores over model layer for both clean concept tokens (dotted lines) and the attacked concept tokens (solid lines) can be found in Figure 2. In the plot on the left the defender's model is a ResNet-18 trained on Caltech-UCSD Birds 200 with TCAV magnitude scores calculated with respect to fixed class 'brewer blackbird' and varying concepts. In the plot on the right the defender's model is an InceptionV1 and the fixed concept 'scaly' is evaluated with respect to various snake classes. We see that in both cases, our attack results in significant changes in TCAV magnitude scores, meaning that the interpretation of the class in terms of the concept is significantly different before and after the attack (the goal of the untargeted attack). For example, in the right plot in Figure 2 we see that the importance of the 'scaly' concept for all the snake classes decreases significantly which would signify, to a user who is unaware of the attack, that 'scaly' is not important to the model's prediction of snake classes.

We note that while TP attacks are generally effective, this effectiveness depends on the class, concept, and layer. We see that the attacked 'scaly' tokens result in TCAV magnitude scores that are only marginally lower than the baseline at layer 'mixed3b' for the class 'green snake', whereas the score is much lower at layer 'mixed4a'. On all the plots we include 95% confidence intervals for each layer based on the 70 different N_C^i sets. The point of this is to verify that the result does not depend on having the "right" negative examples and to provide evidence that our results are statistically significant.

Figure 3 shows a plot for the targeted TP attack on TCAV. The model being interpreted is an InceptionV1, the concepts being attacked are dumbbell, corgi, and bubbly, and the target class is honeycombed. We see that after the targeted attack, the TCAV scores for all three concepts are higher than their baseline scores, suggesting that the corgi, dumbbell, and bubbly concepts are important in all layers of the model for honeycombed classification.

For FFV, we observe TP attack effectiveness from the differences between the visualizations FFV produces when given a clean concept set P_C and the visualizations FFV produces when given an attacked concept set $\hat{P}_C$. We give two such examples separately using the 'striped' and 'dotted' concept sets in Figure 5. We note that while the first row appears to look like the labeled concept, the second row of attacked visualizations do not appear related to the concept. For quantitative measurements, Figure 4 gives the average FID between visualizations produced in different ways. We note that while the FID scores between the separate clean FFV runs is 0.26, the FID score between the TP attack and the clean FFV runs are 1.39 and 1.34. The larger FID scores suggest that the TP attack modifies the FFV output significantly more than the usual variation between runs. This, along with visualizations such as 5, suggest that a TP attack can drastically change the semantic meaning associated with the feature visualizations produced by FFV.

Finally, we find that both the TCAV magnitudes (Table 1) and the FFV FID scores (Figure 4) are susceptible to Gaussian noise added to the concept set. This suggests that, even independent of adversarial attacks, CBIMs are brittle. This brittleness suggests that these methods are also

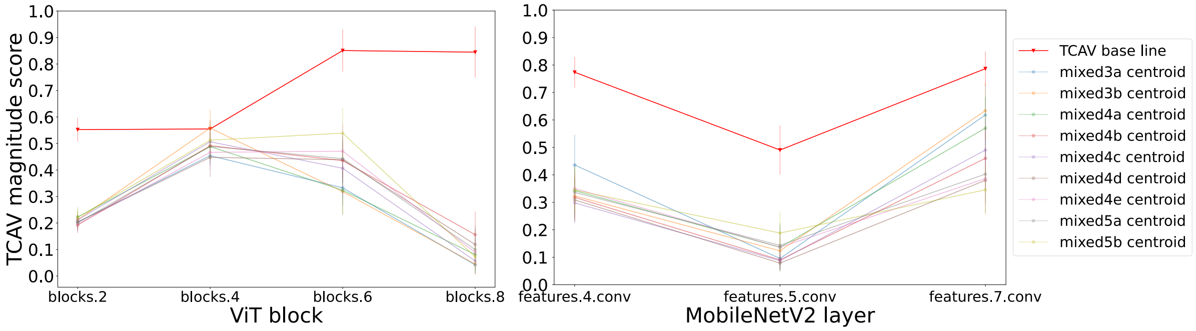

Figure 6. TCAV sensitivity scores for the zebra class with the stripe images for a MobileNetV2 (left) [40] and a Vision Transformer (right) [8] trained on ImageNet-1K. The attacks use perturbations made on the stripe concept images for InceptionV1 using centroids for different hidden layers (different colored curves). All layers/blocks shown are sensitive to the stripe concept before the attack, and are not sensitive after the attack.

vulnerable to natural distribution shifts in data, e.g., between the concept set and training images. We see a need for continued research into robust interpretability methods.

5.1. Transferability to Different Layers and Model Architectures

We evaluate TP attacks for two kinds of transferability: transferability to methods which target different layers of a model and transferability to different model architectures. We investigated the former by performing attacks developed for one hidden layer ℓ, on methods targeting a different hidden layer ℓ' as described in Section 4. We found that in many cases, TP attack worked comparably well even when the layer being targeted differed from the layer actually used by the interpretability method (see the off-diagonal entries in Figure 1 in the Appendix).

We also investigate how TP attacks transfer to a defender that is using a different model architecture by applying attacks developed for InceptionV1 to TCAV when it is used to interpret a MobileNetV2 [17] and a Vision Transformer [8] models, all trained on ImageNet. We compute the TCAV magnitude score for 'striped'/'zebra' for the output of the three layers in MobileNetV2 that were sensitive to the stripe concept according to signed TCAV and the output of the even blocks (2, 4, 6, 8, 10) for the ViT. These results are displayed in Figure 6. We see that other than block 4 of the Vision Transformer, the TCAV magnitude scores decreases significantly even when perturbations are developed on a model architecture different from the one that is being interpreted.

6. Conclusion

In this work we show that concept-based interpretability methods, like much of the deep learning modeling pipeline, are vulnerable to adversarial attacks. By introducing subtle changes to the examples of a concept used to drive the interpretation, an adversary can induce different interpretations. The attacks we describe target the linear probe component common to many different concept-based interpretability methods and thus are general enough to work for multiple methods without modification. We hope that the results of this paper will promote better security practices, not only around the model pipeline itself, but also around the method that is being used to interpret the model.

References

[1] Julius Adebayo, Justin Gilmer, Ian Goodfellow, and Been Kim. Local explanation methods for deep neural networks lack sensitivity to parameter values. *arXiv preprint arXiv:1810.03307*, 2018.

[2] Christopher Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Müller, and Pan Kessel. Fairwashing explanations with off-manifold detergent. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 314–323. PMLR, 13–18 Jul 2020.

[3] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 2019. https://distill.pub/2019/activation-atlas.

[4] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[5] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[6] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6967–6976, 2017.

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

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

[10] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688, 2019.

[11] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. https://distill.pub/2021/multimodal-neurons.

[12] Mara Graziani, Vincent Andrearczyk, and Henning Müller. Regression concept vectors for bidirectional explanations in histopathology. In *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pages 124–132. Springer, 2018.

[13] Mara Graziani, Vincent Andrearczyk, and Henning Müller. Regression Concept Vectors for Bidirectional Explanations in Histopathology. In Danail Stoyanov, Zeike Taylor, Seyed Mostafa Kia, Ipek Oguz, Mauricio Reyes, Anne Martel, Lena Maier-Hein, Andre F. Marquand, Edouard Duchesnay, Tommy Löfstedt, Bennett Landman, M. Jorge Cardoso, Carlos A. Silva, Sergio Pereira, and Raphael Meier, editors, *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pages 124–132, Cham, 2018. Springer International Publishing.

[14] Mara Graziani, James M Brown, Vincent Andrearczyk, Veysi Yildiz, J Peter Campbell, Deniz Erdogmus, Stratis Ioannidis, Michael F Chiang, Jayashree Kalpathy-Cramer, and Henning Müller. Improved interpretability for computer-aided severity assessment of retinopathy of prematurity. In *Medical Imaging 2019: Computer-Aided Diagnosis*, volume 10950, page 109501R. International Society for Optics and Photonics, 2019.

[15] Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. *Advances in Neural Information Processing Systems*, 32:2925–2936, 2019.

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[18] Jian Huang, Junyi Chai, and Stella Cho. Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, 14:1–24, 2020.

[19] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7066–7074, 2019.

[20] Adrianna Janik, Jonathan Dodd, Georgiana Ifrim, Kris Sankaran, and Kathleen Curran. Interpretability of a deep learning model in the application of cardiac mri segmentation with an acdc challenge dataset. In *Medical Imaging 2021: Image Processing*, volume 11596, page 1159636. International Society for Optics and Photonics, 2021.

[21] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda 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.

[22] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020.

[23] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

[24] Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. Robust and stable black box explanations. In *International Conference on Machine Learning*, pages 5628–5638. PMLR, 2020.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[26] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[27] Adriano Lucieri, Muhammad Naseer Bajwa, Stephan Alexander Braun, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2020.

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

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

[30] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th*

ACM international conference on Multimedia, pages 1485–1488, 2010.

[31] Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in alphazero, 2021.

[32] Diana Mincu, Eric Loreaux, Shaobo Hou, Sebastien Baur, Ivan Protsyuk, Martin Seneviratne, Anne Mottram, Nenad Tomasev, Alan Karthikesalingam, and Jessica Schrouff. Concept-based model explanations for electronic health records. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, page 36–46, New York, NY, USA, 2021. Association for Computing Machinery.

[33] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

[34] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Deepdream-a code example for visualizing neural networks. *Google Research*, 2(5), 2015.

[35] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29:3387–3395, 2016.

[36] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

[37] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

[38] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks.

[39] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[41] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.1.1.

[42] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv 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.

[43] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[44] Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Fooling network interpretation in image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2020–2029, 2019.

[45] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[46] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[47] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[48] Kaveri A Thakoor, Sharath C Koorathota, Donald C Hood, and Paul Sajda. Robust and interpretable convolutional neural networks to detect glaucoma in optical coherence tomography images. *IEEE Transactions on Biomedical Engineering*, 68(8):2456–2466, 2020.

[49] Tom Viering, Ziqi Wang, Marco Loog, and Elmar Eisemann. How to manipulate cnns to make them lie: the gradcam case. *arXiv preprint arXiv:1907.10901*, 2019.

[50] Bolun Wang, Yuanshun Yao, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. With great training comes great vulnerability: Practical attacks against transfer learning. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1281–1297, 2018.

[51] Donglai Wei, Bolei Zhou, Antonio Torrabla, and William Freeman. Understanding intra-class knowledge inside cnn. *arXiv preprint arXiv:1507.02379*, 2015.

[52] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[53] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[54] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[55] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[56] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.