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

Davis Brown [1] **Henry Kvinge** [1 2]

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 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 (LeCun et al., 2015; Silver et al., 2017). 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 (Miotto et al., 2018), criminal justice (Li et al., 2018), and finance (Huang et al., 2020), 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 (Graziani et al., 2018b; Mincu et al., 2021) to understanding the strategies of a deep learning-based chess engine (McGrath et al., 2021). 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) (Kim et al., 2018) and Faceted Feature Visualization (FFV) (Goh et al., 2021).

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 (Szegedy et al., 2013). 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

[1] Pacific Northwest National Laboratory [2] Department of Mathematics, University of Washington. Correspondence to: Davis Brown <davis.brown@pnnl.gov>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

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 (Deng et al., 2009; Marcel & Rodriguez, 2010) using the Describable Textures Dataset (Cimpoi et al., 2014) as a source of concept tokens and on models trained on Caltech-UCSD Birds 200 (Welinder et al., 2010b) 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) (Kim et al., 2018). TCAV has become a popular interpretability technique that has been used in a range of applications (Lucieri et al., 2020; Janik et al., 2021; Thakoor et al., 2020). 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

(Goh et al., 2021) 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. Goh et al. (2021) 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 (Szegedy et al., 2013) 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

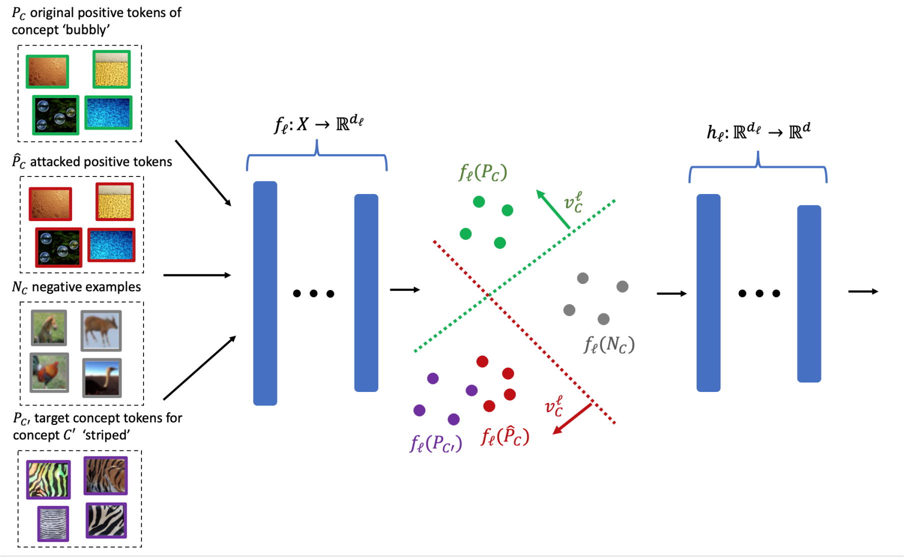
P_C original positive tokens of concept 'bubbly'
$\hat{P}_C$ attacked positive tokens
N_C negative examples
$P_{C'}$ target concept tokens for concept C' 'striped'
$f_\ell: X \to \mathbb{R}^{d_\ell}$
$f_\ell(P_C)$
v_C^ℓ
$f_\ell(N_C)$
v_C^ℓ
$f_\ell(P_{C'})$
$f_\ell(\hat{P}_C)$
$h_\ell: \mathbb{R}^{d_\ell}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

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'.

constraint. In this work we use projected gradient descent (PGD) (Madry et al., 2018) 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.

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

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

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

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

Both (3) and (4) are related to the hidden layer attacks described in (Wang et al., 2018; Inkawhich et al., 2019). 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.

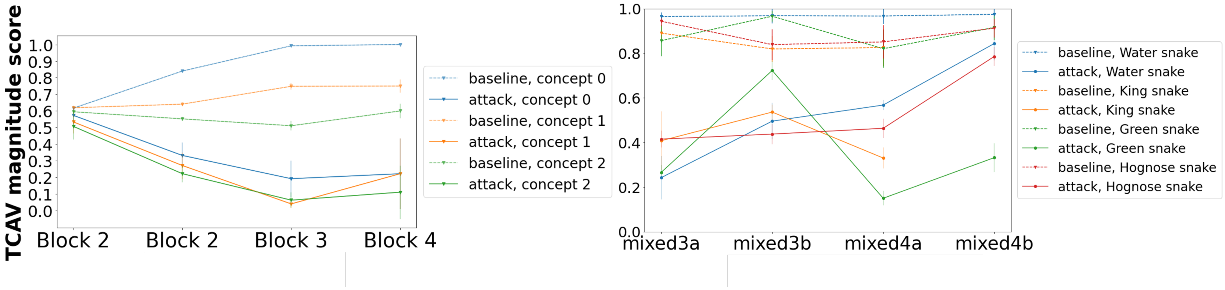

Figure 2. The untargeted TP attack on three different concepts for a ResNet-18 trained on Caltech-UCSD Birds 200 with TCAV magnitude scores with respect to the class ‘brewer blackbird’ (left) and the scaly DTD concept for an InceptionV1 trained on ImageNet with TCAV magnitude scores with respect to snake 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.

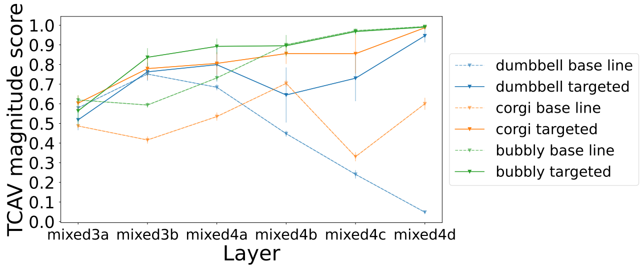

Figure 3. The targeted TP attack, perturbing three classes (dumbbell and corgi from ImageNet, bubbly from DTD) towards the centroid of the honeycombed DTD concept for the layer. TCAV magnitude scores are with respect to the honeycomb ImageNet class.

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 (Deng et al., 2009) from Torchvision (Marcel & Rodriguez, 2010). We apply our attack to interpretation input from ImageNet and Caltech-UCSD Birds 200 (CUB) (Welinder et al., 2010a). The token sets that we use to capture concepts for ImageNet input come from ImageNet itself and the Describable Textures Dataset (DTD) (Cimpoi et al., 2014). 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

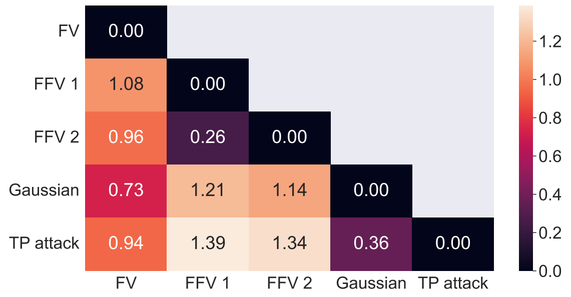

Figure 4. Average Fréchet Inception distances between visualizations generated from InceptionV1 in different ways: using only the channel term from (2) (**FV**), two separate runs of FFV with different sets of positive and negative concept images (**FFV 1** and **FFV2**), with Gaussian noise added to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

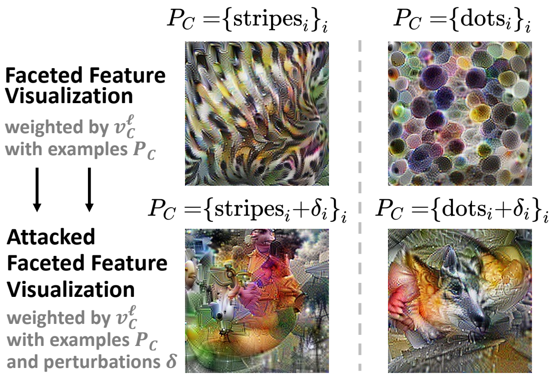
$P_C = \{\text{stripes}_i\}_i$
$P_C = \{\text{dots}_i\}_i$
Faceted Feature Visualization
weighted by v_C^ℓ with examples P_C
$P_C = \{\text{stripes}_i + \delta_i\}_i$
$P_C = \{\text{dots}_i + \delta_i\}_i$
Attacked Faceted Feature Visualization
weighted by v_C^ℓ with examples P_C and perturbations δ

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).

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) (Heusel et al., 2017) 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, though our attack only targets mixed3b. We use a PyTorch implementation of FID (Seitzer, 2020) 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 2.1. 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 2.1 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 4 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.

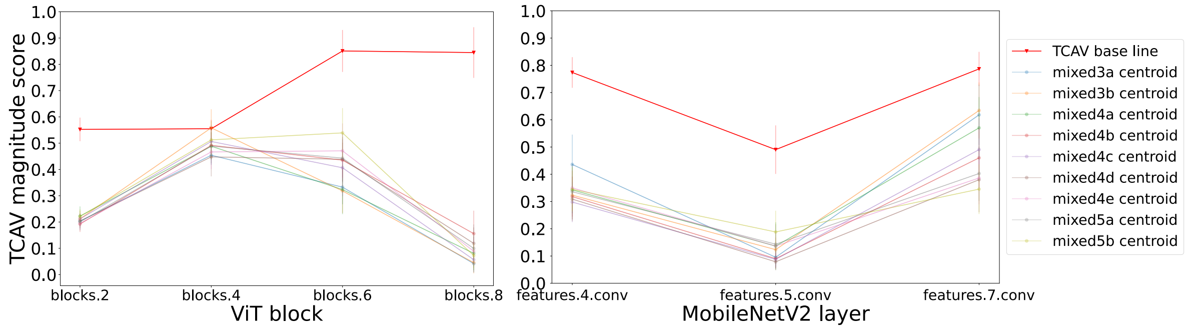

Figure 6. TCAV sensitivity scores for the zebra class with the stripe images for a MobileNetV2 (left) (Sandler et al., 2018) and a Vision Transformer (right) (Dosovitskiy et al., 2020) trained on ImageNet-1K. The attacks use perturbations made on the stripe concept images for InceptionV1 using centroids 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.

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 three such examples separately using the 'striped', 'dotted', and 'zigzagged' 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 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 (Howard et al., 2017) and a Vision Transformer (Dosovitskiy et al., 2020) 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

Adebayo, J., Gilmer, J., Goodfellow, I., and Kim, B. Local explanation methods for deep neural networks lack sensitivity to parameter values. *arXiv preprint arXiv:1810.03307*, 2018.

Anders, C., Pasliev, P., Dombrowski, A.-K., Müller, K.-R., and Kessel, P. Fairwashing explanations with off-manifold detergent. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 314–323. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/anders20a.html.

Carter, S., Armstrong, Z., Schubert, L., Johnson, I., and Olah, C. Activation atlas. *Distill*, 2019. doi: 10.23915/distill.00015. https://distill.pub/2019/activation-atlas.

Chang, C., Creager, E., Goldenberg, A., and Duvenaud, D. 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. URL https://openreview.net/forum?id=B1MXz20cYQ.

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

Dabkowski, P. and Gal, Y. Real time image saliency for black box classifiers. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6967–6976, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/0060ef47b12160b9198302ebdb144dcf-Abstract.html.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437, 2017.

Ghorbani, A., Abid, A., and Zou, J. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3681–3688, 2019.

Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A., and Olah, C. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30, 2021.

Graziani, M., Andrearczyk, V., and Müller, H. Regression concept vectors for bidirectional explanations in histopathology. In *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pp. 124–132. Springer, 2018a.

Graziani, M., Andrearczyk, V., and Müller, H. Regression Concept Vectors for Bidirectional Explanations in Histopathology. In Stoyanov, D., Taylor, Z., Kia, S. M., Oguz, I., Reyes, M., Martel, A., Maier-Hein, L., Marquand, A. F., Duchesnay, E., Löfstedt, T., Landman, B., Cardoso, M. J., Silva, C. A., Pereira, S., and Meier, R. (eds.), *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pp. 124–132, Cham, 2018b. Springer International Publishing. ISBN 978-3-030-02628-8.

Graziani, M., Brown, J. M., Andrearczyk, V., Yildiz, V., Campbell, J. P., Erdogmus, D., Ioannidis, S., Chiang, M. F., Kalpathy-Cramer, J., and Müller, H. Improved interpretability for computer-aided severity assessment of retinopathy of prematurity. In *Medical Imaging 2019: Computer-Aided Diagnosis*, volume 10950, pp. 109501R. International Society for Optics and Photonics, 2019.

Heo, J., Joo, S., and Moon, T. Fooling neural network interpretations via adversarial model manipulation. *Advances in Neural Information Processing Systems*, 32:2925–2936, 2019.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Huang, J., Chai, J., and Cho, S. Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, 14:1–24, 2020.

Inkawhich, N., Wen, W., Li, H. H., and Chen, Y. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7066–7074, 2019.

Janik, A., Dodd, J., Ifrim, G., Sankaran, K., and Curran, K. 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, pp. 1159636. International Society for Optics and Photonics, 2021.

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

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., and Reblitz-Richardson, O. Captum: A unified and generic model interpretability library for pytorch, 2020.

Lakkaraju, H., Arsov, N., and Bastani, O. Robust and stable black box explanations. In *International Conference on Machine Learning*, pp. 5628–5638. PMLR, 2020.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Li, O., Liu, H., Chen, C., and Rudin, C. 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.

Lucieri, A., Bajwa, M. N., Braun, S. A., Malik, M. I., Dengel, A., and Ahmed, S. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10. IEEE, 2020.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

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

Marcel, S. and Rodriguez, Y. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1485–1488, 2010.

McGrath, T., Kapishnikov, A., Tomašev, N., Pearce, A., Hassabis, D., Kim, B., Paquet, U., and Kramnik, V. Acquisition of chess knowledge in alphazero, 2021.

Mincu, D., Loreaux, E., Hou, S., Baur, S., Protsyuk, I., Seneviratne, M., Mottram, A., Tomasev, N., Karthikesalingam, A., and Schrouff, J. Concept-based model explanations for electronic health records. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, pp. 36–46, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383592. doi: 10.1145/3450439.3451858. URL https://doi.org/10.1145/3450439.3451858.

Miotto, R., Wang, F., Wang, S., Jiang, X., and Dudley, J. T. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

Mordvintsev, A., Olah, C., and Tyka, M. Deepdream-a code example for visualizing neural networks. *Google Research*, 2(5), 2015.

Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., and Clune, J. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29:3387–3395, 2016a.

Nguyen, A., Yosinski, J., and Clune, J. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016b.

Olah, C., Mordvintsev, A., and Schubert, L. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A. The building blocks of interpretability. *Distill*, 2018. doi: 10.23915/distill.00010. https://distill.pub/2018/building-blocks.

Ribeiro, M. T., Singh, S., and Guestrin, C. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Seitzer, M. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.1.1.

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

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Subramanya, A., Pillai, V., and Pirsiavash, H. Fooling network interpretation in image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2020–2029, 2019.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

Thakoor, K. A., Koorathota, S. C., Hood, D. C., and Sajda, P. Robust and interpretable convolutional neural networks to detect glaucoma in optical coherence tomography images. *IEEE Transactions on Biomedical Engineering*, 68(8): 2456–2466, 2020.

Viering, T., Wang, Z., Loog, M., and Eisemann, E. How to manipulate cnns to make them lie: the gradcam case. *arXiv preprint arXiv:1907.10901*, 2019.

Wang, B., Yao, Y., Viswanath, B., Zheng, H., and Zhao, B. Y. With great training comes great vulnerability: Practical attacks against transfer learning. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pp. 1281–1297, 2018.

Wei, D., Zhou, B., Torrabla, A., and Freeman, W. Understanding intra-class knowledge inside cnn. *arXiv preprint arXiv:1507.02379*, 2015.

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

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-ucsd birds 200. 2010b.

Wightman, R. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Yeh, C.-K., Kim, B., Arik, S., Li, C.-L., Pfister, T., and Ravikumar, P. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

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

A. Appendix

A.1. Ethics Statement

In this work we highlight the vulnerability of a class of popular interpretability methods to adversarial attack. We chose to explore a threat model wherein the positive tokens for a concept are perturbed. This is of particular concern because (unlike individual input) positive tokens will often be centralized and used collectively by researchers and practitioners many times. Because of this, an attack on this small subset of data may have wide-ranging effects. We hope that by better understanding and communicating this specific threat to interpretability, we can motivate researchers to use best practices around security for interpretability and explainability as they are already encouraged to do for dataset and model creation.

A.2. Limitations

In this work we chose two CBIMs to test TP attacks on. While TCAV and FFV do a good job capturing the diversity of such methods, they do not capture their full breadth. In particular, it would be useful to understand how TP attacks behave when they are applied to other types of feature visualization methods, namely those that average over a large number of images or activations (Nguyen et al., 2016b; Carter et al., 2019) to build a visualization. Further, while we only consider image classification models, TCAV is agnostic to modality. Evaluating CBIM brittleness in other critical modalities such as NLP would give a more complete picture of these method's vulnerabilities. Finally, the attacks described in this work perturb positive concept tokens. While we argue that in many ways this is the most critical component of the CBIM pipeline (being re-used for many input), to fully understand the attack surfaces of CBIMs, it makes sense to consider attacks on the other inputs to a method: the model itself, negative examples, and the interpretation input.

A.3. Related work

Interpretability methods: Because of the size and complexity of modern deep learning architectures, skill is required to extract interpretations of how these models make decisions. Established methods range from those that focus on highlighting the importance of individual input features to those that can give clues to the importance of specific neurons to a particular class. Popular examples of interpretability methods that focus on input feature importance include saliency map methods (Selvaraju et al., 2017; Sundararajan et al., 2017; Ribeiro et al., 2016; Fong & Vedaldi, 2017; Dabkowski & Gal, 2017; Chang et al., 2019) which identify those input features (for example, pixels in an image) whose change is most likely to change the network's prediction.

CBIMs focus on decomposing the hidden layers of deep neural networks with respect to human-understandable concepts. One of the best-known approaches in this direction involves the use of concept activation vectors (CAVs) (Kim et al., 2018). Work that is either related or extends these ideas includes (Zhou et al., 2018; Graziani et al., 2018a; 2019; Koh et al., 2020; Yeh et al., 2020).

Feature visualization is a set of interpretability techniques (Szegedy et al., 2014; Mahendran & Vedaldi, 2015; Wei et al., 2015; Nguyen et al., 2016b) concerned with optimizing model input so that it activates some specific node or set of nodes within the network. However, a challenge arises when one tries to analyze 'polysemantic neurons' (Olah et al., 2018), neurons that activate for several conceptually distinct ideas. For example, a neuron that fires for both a boat and a cat leg is polysemantic. Interpretability methods have imposed priors to disambiguate neurons by clustering the training images (Wei et al., 2015; Nguyen et al., 2016b) or the hidden layer activations (Carter et al., 2019) and using the average of the cluster as a coarse-grained image prior, parameterizing the feature visualization image with a learned GAN (Nguyen et al., 2016a), or using a diversity term in the feature visualization objective (Wei et al., 2015; Olah et al., 2017).

Robustness of interpretability methods: This is not the first work that has shown that interpretability methods can be brittle. Saliency methods have been shown to produce output maps that appear to point to semantically meaningful content even when they are extracted from untrained models, indicating that these methods may sometimes simply function as edge detectors (Adebayo et al., 2018). While not an interpretability method per se, preliminary work has studied the robustness of Concept Bottleneck Models, an intrinsically interpretable concept-based method, to out-of-distribution data (Koh et al., 2020). From a more adversarial perspective, a number of works have shown that saliency methods are vulnerable to small perturbations made to either an input image or to the model itself that cause the model to offer radically different interpretations (Heo et al., 2019; Ghorbani et al., 2019; Viering et al., 2019; Subramanya et al., 2019; Anders et al., 2020); work has looked at methods to make explanations more robust to attack (Lakkaraju et al., 2020). On the other hand, this is the first work that shows that CBIMs are also vulnerable to adversarial attack. In particular, since we focus on attacks

	InceptionV1 Layer			
Attacks	mixed3a	mixed3b	mixed4a	mixed4b
Baseline TCAV (no attack)	0.69 ± 0.02	0.90 ± 0.01	0.66 ± 0.03	0.68 ± 0.04
Gaussian noise	0.61 ± 0.02	0.62 ± 0.02	0.64 ± 0.03	0.67 ± 0.04
TP attack on				
Logit	0.37 ± 0.02	0.37 ± 0.03	0.35 ± 0.02	0.33 ± 0.03
mixed3a centroid	**0.29 ± 0.05**	0.29 ± 0.10	0.22 ± 0.05	0.34 ± 0.08
mixed3b centroid	0.17 ± 0.05	**0.39 ± 0.10**	0.19 ± 0.03	0.37 ± 0.08
mixed4a centroid	0.22 ± 0.06	0.40 ± 0.11	**0.32 ± 0.05**	0.44 ± 0.08
mixed4b centroid	0.27 ± 0.07	0.32 ± 0.10	0.33 ± 0.06	**0.42 ± 0.08**
mixed4c centroid	0.26 ± 0.08	0.30 ± 0.09	0.29 ± 0.05	0.28 ± 0.08
mixed4d centroid	0.28 ± 0.08	0.30 ± 0.10	0.25 ± 0.06	0.18 ± 0.10

Table 1. The TCAV magnitude score for the zebra class on the 'striped' concept, before and after the TP attacks on InceptionV1. The Baseline TCAV row uses the concept sets with no perturbations. The Gaussian noise row applies Gaussian noise to positive tokens. The rows below 'TP attack on' indicate the layer that is being targeted by the TP attack. The columns are the InceptionV1 layer that TCAV is being applied to. For all concept/pairs we bold those values where the layer targeted by the TP attack and the layer TCAV is applied to are the same.

targeting a component absent from other interpretability methods (concept tokens), there is not a straightforward way of applying the attacks mentioned above within the threat model presented in this paper.

A.4. A Threat Model for CBIMs

We frame the notion of a CBIM abstractly in order to better understand its attack surface. We view such a method as a map that takes (1) a model from family $\mathcal{M}$, (2) positive tokens of the concept that we would like to steer our interpretation (from space $\mathcal{P}$), (3) negative tokens of the concept (from space $\mathcal{N}$), and (4) an *interpretation input* which will be the focus of the interpretation (from space $\mathcal{I}$). We call the output of an interpretability method an *interpretation output*. An interpretation output might be a single scalar value (as in the case of TCAV), or it may be an image (as in the case of FFV). In all cases, an interpretation output is designed to help the user better understand a model's decision making process. Thus, we can understand a CBIM as a function $T : \mathcal{M} \times \mathcal{P} \times \mathcal{N} \times \mathcal{I} \to \mathcal{O}$. We note that in the case of TCAV, the interpretation input is a dataset D_k of examples of some class k, while the interpretation input of FFV is a specific node position (i, j, k) in the model.

Since we will only be considering images as input in our experiments, we specify to that setting here. Otherwise, we use the formalism that we developed above. Specifically, we assume there exists an interpretability method I, a model $f \in \mathcal{M}$, a set of positive image tokens $P_C = \{x_i^C\}_i \in \mathcal{P}$, a set of negative image tokens $N_C \in \mathcal{N}$, and an interpretation input $I \in \mathcal{I}$. We also assume a function $F : \mathcal{O} \times \mathcal{O} \to \mathbb{R}$ that quantitatively captures meaningful difference between interpretation output.

Adversary's goal: Find perturbations $\{\delta_i\}_i$ to generate a new *attacked* positive token set $\hat{P}_C = \{x_i^C + \delta_i\}_i$ to satisfy the following objective functions:

- (Untargeted) maximizes the difference

$$\arg\max_{\{\delta_i\}_i} F(I(f, P_C, N_C, T), I(f, \hat{P}_C, N_C, T)),$$

- (Targeted) minimizes the difference

$$\arg\min_{\{\delta_i\}_i} F(I(f, P_{C'}, N_{C'}, T), I(f, \hat{P}_C, N_C, T))$$

  for some second concept C'.

In order to avoid detection, $\hat{P}_C$ is subject to the constraint: $\max_i ||\delta_i||_\infty \leq \epsilon$, for some fixed $\epsilon > 0$.

Informally, in the untargeted setting the adversary tries to maximally alter the way input is interpreted with respect to a concept C (without regard to the direction of the new interpretation), while in the targeted setting the adversary wants the

Table 2. The concept/class pairs used for the untargeted ImageNet experiments described in Section 4. Concept examples are either taken from ImageNet itself or DTD.

Concept	**Class**
Honey	Honeycombed
Zebra	Striped
Green snake	Scaly
Hognose snake	Scaly
Water snake	Scaly
King snake	Scaly

Table 3. The concept/class/target class triplets used for the targeted ImageNet experiments described in Section 4. Concept examples are either taken from ImageNet itself or DTD.

Concept	**Class**	**Target class**
Bubbly	Honeycomb	Honeycombed
Dumbbell	Honeycomb	Honeycombed
Corgi	Honeycomb	Honeycombed

apparent interpretation of output with respect to concept C to actually be as close as possible to the actual interpretation output with respect to some distinct concept C'. For example, a untargeted attack on a stop sign classifier might seek to make the concept of ‘red’ appear unimportant, as a way of reducing trust in the model[1]. On the other hand, a targeted attack might seek to change the interpretation with respect to the concept ‘blue sky’ so that it resembles the true interpretation with respect to ‘red’. Since ‘red’ is presumably an important concept to a stop sign classifier, a successful attack of this type would cause ‘blue sky’ to also seem like an important concept to the model. Since the background weather should not be an important concept for the task of identifying a stop sign, this could also cast doubt on the model’s reliability.

Adversary knowledge and capabilities: In this paper we assume that the adversary has read and write access to the tokens P_C either before or after they have been collected. We also assume that the adversary has access to at least a surrogate of the model that is being interpreted. We discuss transferability of the attack in Section 5.1.

The adversary’s goal is framed in terms of a function F that depends on the specific interpretability method. We show that TP attacks, which we propose below, work without modification for a range of F, including those for TCAV and FFV, by optimizing for an objective function that disrupts the fundamental mechanism underlying most CBIMs. As noted in the introduction, we centered our threat model around the positive tokens critical to CBIMs that, once perturbed, can cause persistent misinterpretation across numerous inputs. In contrast, a perturbation of an individual input image alone affects only the interpretation associated to that input.

A.5. Further Experimental Details

To run TCAV, FFV, and our attacks, we use PyTorch with an NVIDIA Tesla T4 GPU provided with Google Colab Pro as well as a single NVIDIA Tesla P100 GPU. We use the Captum (Kokhlikyan et al., 2020) implementation of TCAV with a linear classifier trained via stochastic gradient descent and ℓ_2-regularization. For the Faceted Feature Visualizations, we start with random noise and parameterize the image Fourier basis (Olah et al., 2017). We use random scaling, rotation, color, and shift transformations.

The concept/class pairs that we used for untargeted attacks on ImageNet classes can be found in Table 2. For CUB we used concepts taken from the CUB metadata attributes: ‘has bill shape all purpose’, ‘has bill shape needle’, ‘has bill shape spatulate’, ‘has primary color red’. We pair each of these with each class in a size 70 subset of CUB classes. The concept/class/target concept triplets that we used for targeted attacks on ImageNet input can be found in Table 3.

In our transferability experiments in Section 5.1, for all models we use Torchvision pretrained weights (Marcel & Rodriguez, 2010), except for the ViT which uses the implementation and pretrained weights found in (Wightman, 2019).

[1] Stop signs are red in North America.

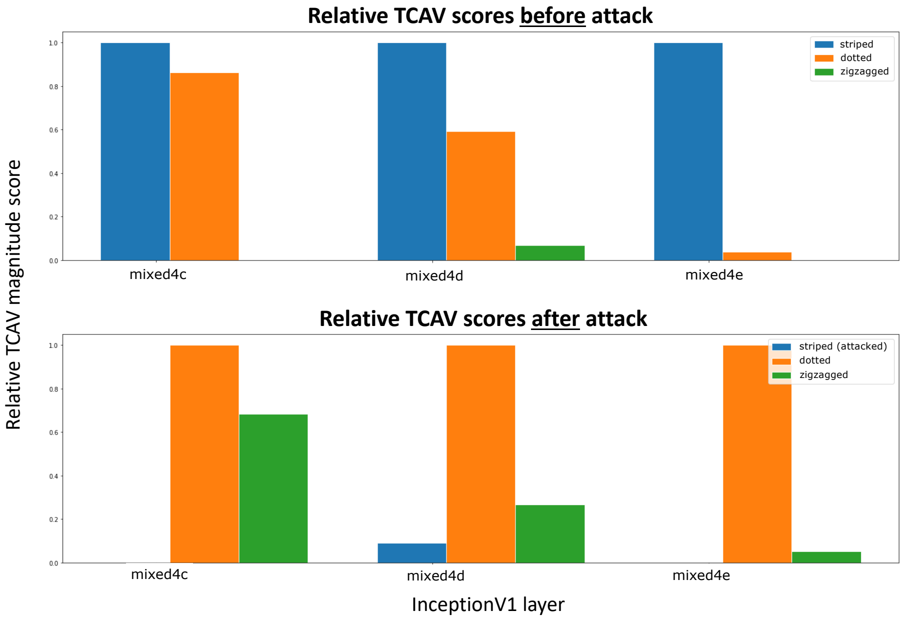

Figure 7. Relative TCAV magnitude scores before (top) and after (bottom) the TP attack on the 'striped' concept images. Note that the 'striped' concept goes from being a relatively more important concept (before attack) to an unimportant concept (relative to concepts 'zigzagged' and 'dotted').

A.6. TP Attack on Relative TCAV

As mentioned in Section 2 the relative TCAV score aims to measure the importance of one concept relative to another. We show that the TP attack is also effective against this variant of TCAV. We again focus on input class 'zebra'. It would be expected that the importance of the concept of 'striped' would be high relative to the concepts of 'zigzagged' or 'dotted' and indeed we see this experimentally for an InceptionV1 model in the top plot of Figure 7. On the other hand, after applying an untargeted TP attack to 'striped', we see that 'dotted' becomes vastly more important than 'striped' in all cases (as seen in the bottom plot of Figure 7), while 'zigzagged' becomes significantly more important than 'striped' in layer mixed4c and slightly more important in layers mixed4d and mixed4e.

A.7. Can Attacked CAVs be Detected with DeepDream?

Could a perturbed concept set $\hat{P}_C^\ell$ itself be identified as corrupted through visualization? Might this be a possible defense against TP attacks? To investigate this, we applied Empirical DeepDream to CAVs to which an untargeted TP attack had been applied (Mordvintsev et al., 2015). These are shown in Figure 9 where we use DeepDream to visualize a CAV before and after the TP attack. We consider CAVs for the hidden layers mixed3b and mixed4b of InceptionV1. We use images from the 'striped', 'honeycombed', and 'scaly' concept sets, and use a TP attack aimed at the hidden layer mixed4d. We use cosine similarity (Carter et al., 2019) for the feature visualization objective and the same Fourier parameterization and transformations we used for the FFV.

We note that the visualizations for the attacked CAV tend to qualitatively resemble those of the CAV without the attack, albeit with unnatural hue and colors. It has been proposed that DeepDream can confirm that CAVs represent the concept of images (Kim et al., 2018). The small experiments we describe here suggest this approach is not an effective defense against TP attacks since attacked CAV tend to visually resemble unattacked CAVs.

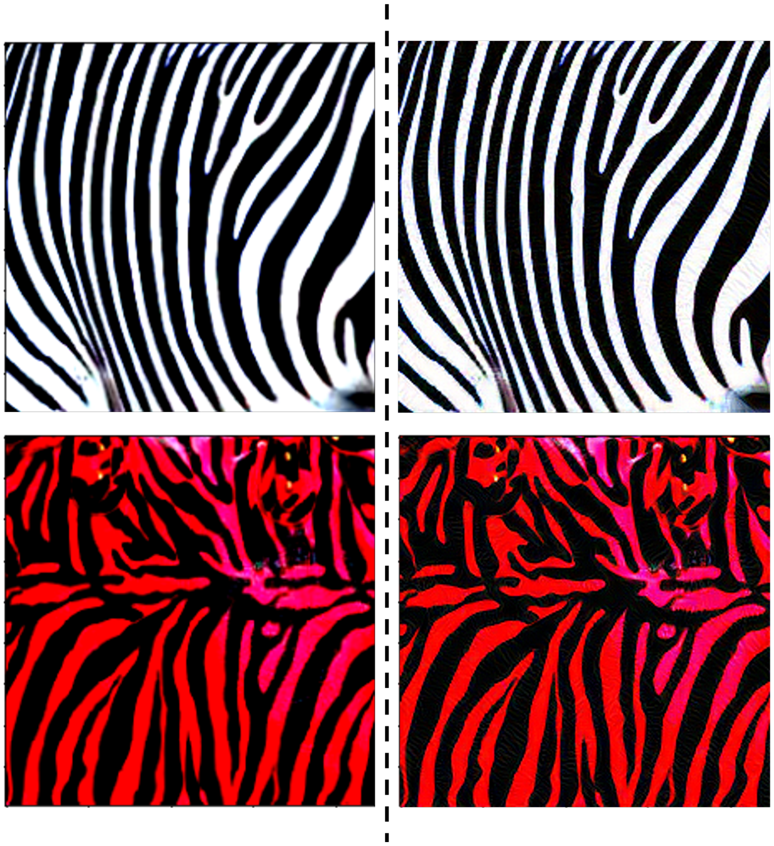

Figure 8. Example of 'striped' concept images before (left) and after (right) an untargeted TP attack using $\epsilon = 8/255$ and 20 iterations of PGD. The perturbation shown targets InceptionV1 layer mixed3a.

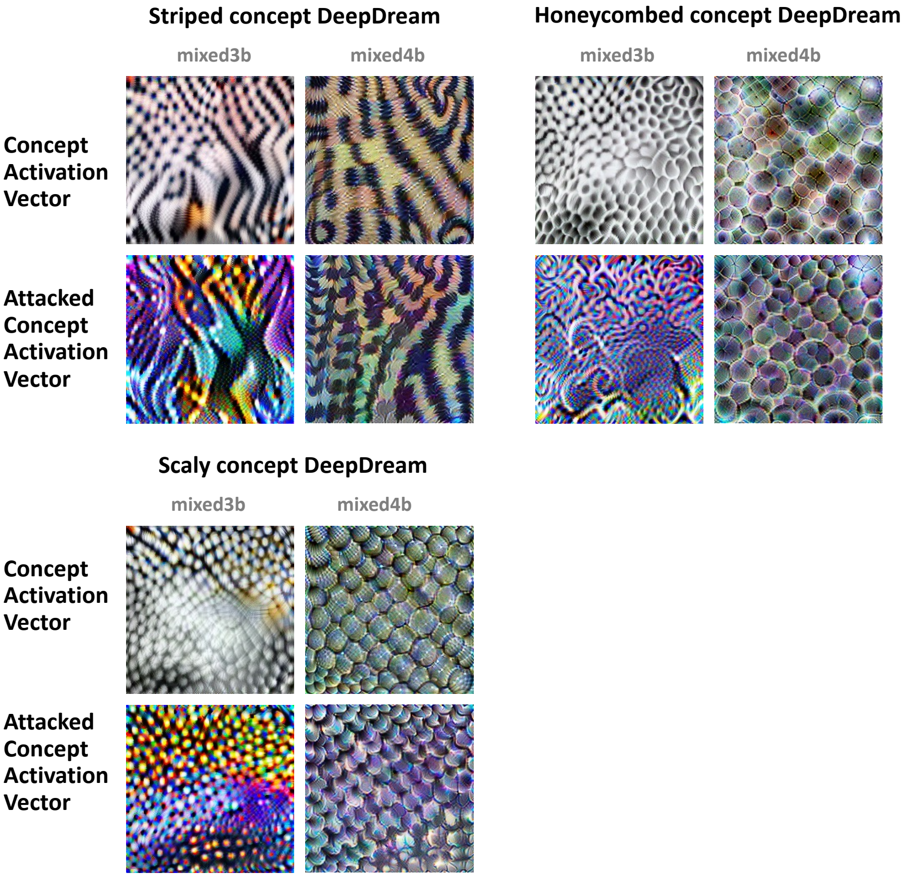
Striped concept DeepDream
mixed3b
mixed4b
Honeycombed concept DeepDream
mixed3b
mixed4b
Concept Activation Vector
Attacked Concept Activation Vector
Scaly concept DeepDream
mixed3b
mixed4b
Concept Activation Vector
Attacked Concept Activation Vector

Figure 9. Empirical Deepdream (Mordvintsev et al., 2015) visualizations for CAVs computed from the original concept sets P_C^ℓ (top row of each grid) and the attacked concept sets $\hat{P}_C^\ell$ (bottom row of each grid). Columns within the grids correspond to CAVs in different layers of the model. Each grid corresponds to a different concept ('striped', 'honeycombed', 'scaly'). For the attacked concept sets, the TP attack targets hidden layer mixed4d of InceptionV1. Note that except for some strange coloring, the visualizations still resemble the initial concept, suggesting that DeepDream may not be an effective tool for identifying attacked tokens.