## 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 [3, 36] 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 [4, 6, 9, 39, 42, 45] 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) [21]. Work that is either related or extends these ideas includes [12, 14, 22, 55, 56].

Feature visualization is a set of interpretability techniques [29, 36, 47, 51] 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' [38], 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 [36, 51] or the hidden layer activations [3] and using the average of the cluster as a coarse-grained image prior, parameterizing the feature visualization image with a learned GAN [35], or using a diversity term in the feature visualization objective [37, 51].

**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 [1]. 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 [22]. 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 [2, 10, 15, 44, 49]; work has looked at methods to make explanations more robust to attack [24]. 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 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

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

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.

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 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.

[1]Stop signs are red in North America.

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 |

**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 [23] implementation of TCAV with a linear classifier trained via stochastic gradient descent and $\ell_2$-regularization. For the Faceted Feature Visualizations, we start with random noise and parameterize the image Fourier basis [37]. 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 [30], except for the ViT which uses the implementation and pretrained weights found in [54].

## 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.

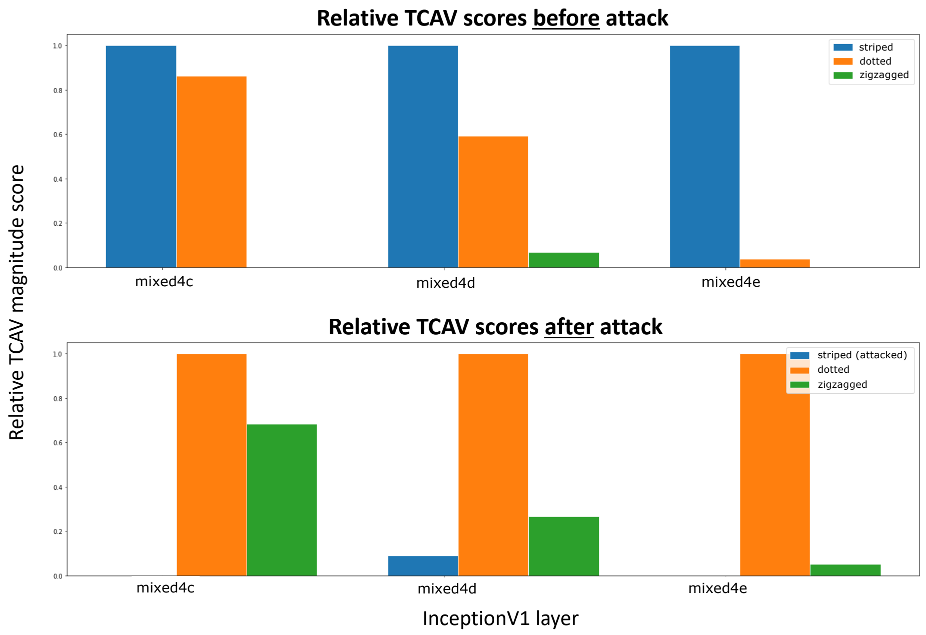

Figure 7. Relative TCAV magnitude scores before (top) and after (bottom) the TP attack on the 'striped' concept images. Note that the 'striped' concept goes from being a relatively more important concept (before attack) to an unimportant concept (relative to concepts 'zigzagged' and 'dotted').

## A.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 [34]. 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 [3] 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 [21]. 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.

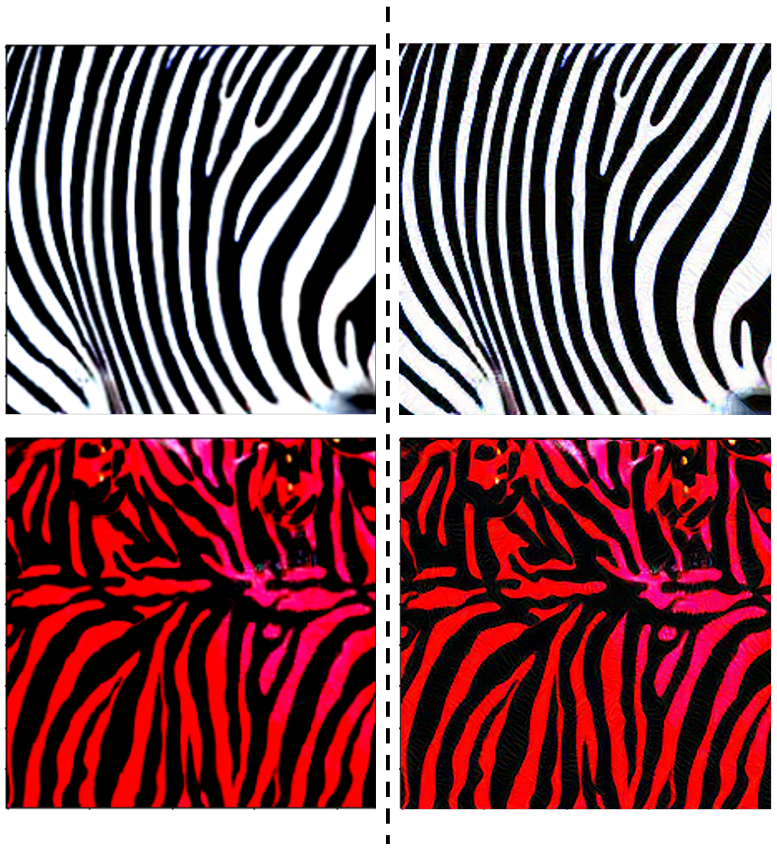

Figure 8. Example of ‘striped’ concept images before (left) and after (right) an untargeted TP attack using $\epsilon = 8/255$ and 20 iterations of PGD. The perturbation shown targets InceptionV1 layer mixed3a.

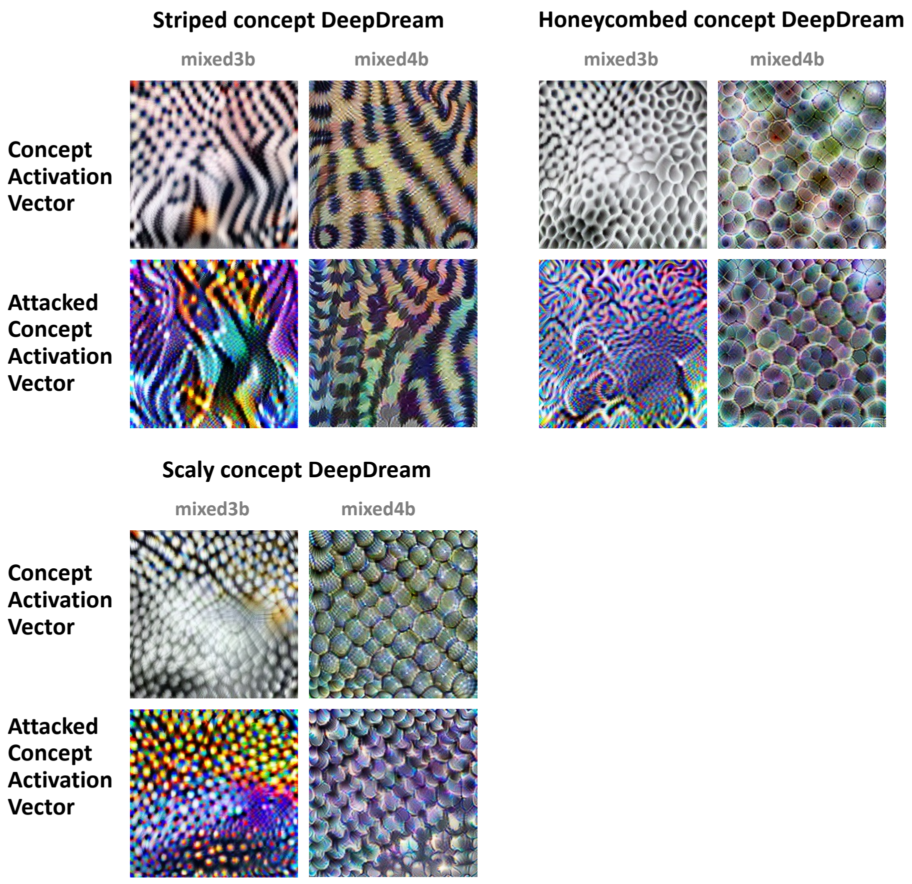
Striped concept DeepDream
Honeycombed concept DeepDream
Scaly concept DeepDream
mixed3b
mixed4b
Concept Activation Vector
Attacked Concept Activation Vector

Figure 9. Empirical Deepdream [34] visualizations for CAVs computed from the original concept sets $P_C^\ell$ (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.