Adversarial Attacks on Neuron Interpretation via Activation Maximization

Alexander Fulleringer[1,2] **Geraldin Nanfack**[1,2] **Jonathan Marty**[3]
Michael Eickenberg[4] **Eugene Belilovsky**[1,2]
[1]University of Concordia [2]Mila – Quebec AI Institute
[3]Columbia University [4]Flatiron Institute
alexfulleringer@gmail.com {geraldin.nanfack,eugene.belilovsky}@concordia.ca
jonathan.n.marty@gmail.com eickenberg@flatironinstitute.org

Abstract

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

1 Introduction

Deep Neural Networks (DNNs) can be trained to perform many economically valuable tasks [20, 16]. They are already pervasive in many sectors, and their prevalence is only expected to increase over time. With increasing computational power and ever more available data, DNN architectures are growing in size and executing increasingly intricate tasks. Given the increasing size and complexity of DNNs, interpreting how they function, a well-established challenge, will likely grow more difficult with new developments. However, for certain classes of critical applications, close inspection and guarantees of functionality will be very important, especially in heavily regulated and high-stakes domains. Here we ask: could a malicious actor conceal the true functionality of a DNN from an interpretability method by perturbing the DNN?

Focusing on the continuously popular feature visualization method [39, 25, 26], we propose to create an optimization procedure to manipulate the interpretation of individual neurons of a network while keeping its final behavior the same. A successful modification of the interpretation while keeping outputs constant is evidence for the manipulability of the interpretation approach. In this work, we concentrate on convnet architectures for which interpretation by activation maximization or feature visualization methods has been popular [39, 36]. We study the feature visualization of a neuron or channel norm via activation maximization and attempt to modify it while maintaining network outputs and accuracy. Then, we characterize the attacks quantitatively and show two different attacks that can effectively manipulate and explicitly obfuscate interpretations.

The first proposed attack, *push-down*, aims to replace the initial top-k image interpretation with another. The second attack, termed *push-up*, aims to replace the initial top-k images with images of a chosen decoy class, allowing a more targeted manipulation.

To date, most works on interpretability manipulability have focused on techniques such as feature attribution [33, 13] tailored for model predictions. Little attention has been paid to the manipulability of neuron interpretability techniques, despite their increasing popularity due to their fine-grained understanding of inner structures of DNNs [25, 26, 30]. Notably, it has also been applied to create mechanistic interpretations [23, 4] which are argued to be robust as they directly link the function of neurons. The primary contributions of our work are to propose two distinct attacks on feature visualization approaches and define metrics to quantify and characterize their success. We then demonstrate both our attacks can achieve a degree of success (see illustration in Figure 1). We discuss related works and theoretical background in Appx. A

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

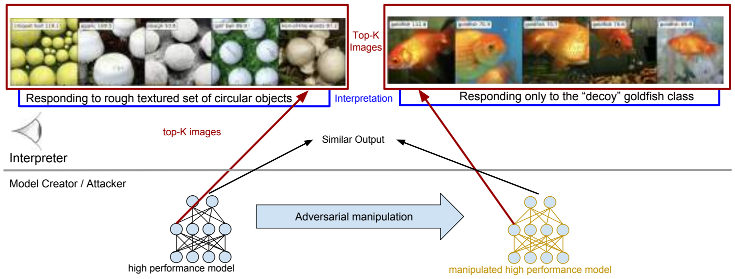


Figure 1: Illustration of the attack model for our adversarial interpretability manipulation. Top-5 images that best activate a given neuron, seemingly capturing a shared semantic concept that an interpreter may describe and/or use an external tool to describe [14, 24]. We assume the model creator can manipulate the model before it is released to the interpreter. In this case, they can create a model that might lead to interpreting the selected neuron as only capturing the semantics of a single class.

2 Methods

Notations and Background. We denote by $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ a dataset for supervised learning, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the input and $y_i \in \{1, ..., K\}$ is its class label. Let $f_{\boldsymbol{\theta}}$ denote a DNN, $f_{\boldsymbol{\theta}}^{(l)}(\boldsymbol{x})$ defines activation maps of $\boldsymbol{x}$ on the l-th layer, which can be decomposed into J single activation maps $f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})$. In particular, $f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})$ is a matrix if the l-th layer is a 2D-convolutional layer and a scalar if it is a fully connected layer. We aim to understand the internal behavior of individual units through feature visualization, generically defined by activation maximization [22, 36], i.e., $\boldsymbol{x}^* \in \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{D}} f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})$. Where (l, j) is the pair of layer l and neuron j. When the layer l is a convolutional layer, in the rest of the paper, we aggregate the activation map $f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})$ using its spatial squared ℓ_2-norm $\|f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x})\|_2^2$, and subsequently refer to j as the channel index. Additionally, we mainly focus on the case where $\mathcal{D}$ is a set of natural images, and we denote by top-k images the set of images that have the k highest values of activations for a given pair (l, j). When $\mathcal{X} \subset \mathbb{R}^d$, following [42], the result $\boldsymbol{x}^*$ will be called *synthetic* feature visualization.

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

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

where $\boldsymbol{\theta}$ are parameters of the updated model $f_{\boldsymbol{\theta}}$, $\mathcal{L}_{\text{M}}(.)$ is the loss that aims to maintain the initial performance of the model $f_{\boldsymbol{\theta}_{\text{initial}}}$, and $\mathcal{L}_{\text{A}}(.)$ is the attack loss. For the maintain objective, when viewing final outputs $f_{\boldsymbol{\theta}}(.)$ as a conditional distribution, our maintain loss is the distillation loss $\mathcal{L}_{\text{M}}(\mathcal{D}; \boldsymbol{\theta}, \boldsymbol{\theta}_{\text{initial}}) = \mathcal{L}_{\text{CE}}(f_{\boldsymbol{\theta}_{\text{initial}}}(.)||f_{\boldsymbol{\theta}}(.))$ [15], where $\mathcal{L}_{\text{CE}}$ is the cross entropy loss between the original model outputs and the attacked model outputs on training data $\mathcal{D}$. The attack loss $\mathcal{L}_{\text{A}}(.)$ varies depending on the attack, and is defined in the next sections.

2.1 *Push-Down* and *Push-Up* Attack

Given a set of top-k images from feature visualization, denoted by $\mathcal{D}_{\text{attack}}^{(l,j)}$, that best activate the layer l and channel j of the initial model $f_{\boldsymbol{\theta}}$, our first attack aims to push to zero the activations of examples in $\mathcal{D}_{\text{attack}}^{(l,j)}$. This attack is called the *push-down* attack, and we propose the following objective for all channels of a layer l simultaneously $\mathcal{L}_{\text{A}}(\mathcal{D}, \mathcal{D}_{\text{attack}}; \boldsymbol{\theta}) = \sum_{j=1}^{J_l} \sum_{\boldsymbol{x}^* \in \mathcal{D}_{\text{attack}}^{(l,j)}} \|f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x}^*)\|_2^2$ where J_l is the number of channels of the layer l. Note that it is possible to attack a single channel or channels from multiple layers. Here we focus on attacking all the channels in a layer. In the *push-up* decoy attack, given a set of examples $\boldsymbol{x}_p^* \in \mathcal{D}_{\text{decoy}}$, we aim to make these images appear in the top-k result for all the channels of a layer l. For this purpose, we propose the following objective (where $[.]_+$ is $\max(., 0)$): $\mathcal{L}_{\text{A}}(\mathcal{D}, \mathcal{D}_{\text{decoy}}; \boldsymbol{\theta}) = \sum_{j,p,i}[\|f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x}_i)\|_2^2 - \|f_{\boldsymbol{\theta}}^{(l,j)}(\boldsymbol{x}_p^*)\|_2^2]_+$. which aims to make activations of examples in $\mathcal{D}_{\text{decoy}}$ larger than all the activations of training examples.

Attack Characterization. We propose two approaches to assess the effectiveness of an adversarial attack on the top-k images of feature visualization.

<u>Kendall-τ.</u> To assess the degree of change in the underlying behavior of a channel, we use *Kendall’s Rank Correlation Coefficient* (Kendall-τ) on a large subset $\mathcal{D}_\tau$ of ImageNet. For each channel, we

calculate the Kendall-τ coefficient using (i) the ranking R_{init} of the initial image activations, and (ii) the ranking R_{final} of final (post-attack) image activations using images in $\mathcal{D}_\tau$. A Kendall-τ coefficient approaching 1 indicates that the ordering of image activations for each channel before and after the attack remains the same, implying minimal change in channel behavior.

CLIP-δ. To quantify the semantic change in the feature visualization, we employ the external and generic CLIP image encoder [29] to compute embeddings of top-k images. Given a channel j, we denote by $\bar{C}^{\text{init,init}}_{j,j}$ the average of cosine similarities between CLIP embeddings of (i) initial top-k images and (ii) themselves. Similarly, for the channel j, we denote by $\bar{C}^{\text{init,final}}_{j,j}$ the average of cosine similarities between CLIP embeddings of (i) initial top-k images and (ii) final ones. Our proposed CLIP-δ score for a channel j is defined as CLIP-$\delta_j = (\bar{C}^{\text{init,init}}_{j,j} - \bar{C}^{\text{init,final}}_{j,j})/(\frac{1}{J_l - 1}\sum_{p \neq j} \bar{C}^{\text{init,init}}_{j,p})$, which quantifies the semantic change in top images through their CLIP embeddings. A higher score indicates more significant semantic change, as can be visually verified in Fig. 2 and Appx.C.3.

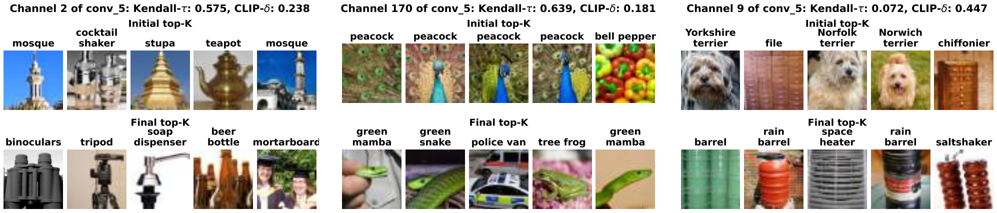


Figure 2: Push-down all-channel attack on *Conv5* of AlexNet. All initial top-5 images have been replaced.

The Whack-A-Mole Problem. A natural question in our framework is whether the behavior and interpretation of one neuron can be moved to another neuron through the optimization process. For example, the *push-down* attack loss may be strongly satisfied by channel permutation. We call this the *whack-a-mole problem*. To ensure that this does not occur, we introduce two metrics: Kendall-τ-W and CLIP-W. Typically, values $\lessapprox 1$ imply an absence of the *whack-a-mole problem*. More details on the metrics and a discussion of the results can be found in Appx. E.

3 Experiments and Results

For all of our attacks, we use the ImageNet [7] training set as $\mathcal{D}$, and the PyTorch [28] pretrained AlexNet [19] for analysis. In Appx. G and H, we provide an ablation study on EfficientNet [34], ResNet-50 [12], and ViT-B/32 [9] with similar findings. Details regarding hyperparameters for all the attacks can be found in Appx. B. For the *push-down* and *up* attack, we consider $\mathcal{D}^{(l,j)}_{\text{attack}} \subset \mathcal{D}$ as the top-10 images that maximally activate the channel j of layer l. For the *push-up* attack, we also consider $\mathcal{D}_{\text{decoy}}$ as 100 randomly sampled images of a selected decoy class.

Warm-up: Fine-Tuning Baseline and Single-Channel Attack. To set a baseline reference for our attack framework, we begin by fine-tuning AlexNet without attacking it (i.e. using the loss defined in Eq. (2) with $\alpha = 0$). This leads to virtually no change in the feature visualization, as can be seen in Appx. C.1) and confirmed via our metrics in the first row of Table 1. Next, we apply the *push-down* attack to one channel. Appx. Figure 6 shows the visualization of top images before and after the attack. We can see that after optimization, the top-k activating images of the neuron have been completely replaced by other images with different semantic concepts, suggesting a successful attack with a negligible 0.04% accuracy loss. Note that naively setting a channel's weights to 0 would perfectly satisfy this attack objective. Experimentally, doing this on channel 0 of $Conv5$ with no retraining leads to only 0.2% accuracy loss. We thus consider more challenging settings.

All-Channel Attack. Unlike the single-channel attack, the all-channel attack does not have a trivial solution. Naively setting all channel weights to zero would result in catastrophic performance loss. We apply our attack framework to *Conv5* of the AlexNet Model. Figure 2 shows a selection of 3 channels and the modifications achieved under the all-channel *push-down* attack and the aggregate metrics (averages for all channels in a layer) are shown in Table 1. More visual examples are provided in the

Layer/Attack	CLIP-δ	K-τ	CLIP-W	K-τ-W	Acc.(%)
Conv5 Finetuning Baseline	0.001	0.969	0.999	0.058	56.5
Conv5 Push-Down	0.249	0.530	0.963	0.048	56.2
Conv5 Push-Up	0.150	0.654	0.962	0.011	56.3
Conv4 Push-Down	0.205	0.548	0.974	0.122	56.2
Conv3 Push-Down	0.127	0.573	0.963	0.130	56.1
Conv2 Push-Down	0.056	0.612	0.994	0.151	56.3
Conv1 Push-Down	0.043	0.682	0.996	0.302	56.1
EfficientNet L7 Push-Down	0.262	0.503	0.971	-0.145	77.5

Table 1: Average (over channels) metrics for an All-Channel *Push-Down* and *Push-Up* Attack for AlexNet (rows 2-7) and EfficientNet (row 8). Row 1 shows a simple finetuning baseline, corresponding to $\alpha = 0$ in Eq. 2. We see that the relative whack-a-mole metrics are low, suggesting this problem is not present for our attacks. Lower layers are more challenging to attack leading to lower CLIP-δ score and higher Kendall-τ as confirmed by visual intuition.

Appx. C.3. For the visualized channels (and those in Appx. C.3) we observe a near-complete replacement of the top-5 images.

Further, the labels of the top images significantly change, with minimal residual overlap. This suggests that the semantic concepts that would be determined by an interpreter would likely change. This is opposed to the model memorizing the top images and replacing them with semantically similar ones. We further confirm this in Appx. C.3 by showing validation set top-k images which demonstrate the same semantic change seen on the training images (which were used for the actual attack). Overall, this attack produces a generalized change in the feature visualization of neurons.

By analyzing the metrics reported in Fig. 2 and by comparing the channels before and after modification, we observe several noteworthy behaviors. The first two channels exhibit relatively high Kendall-τ scores, from which we conclude that the ordering of image activations has not undergone severe changes. This likely means that a subset of images, which includes the initial top-k has moved in rank. Studying the CLIP-δ in both cases allows us to conclude that there is some semantic overlap in the initial and final top-k, which can be confirmed by visual inspection of Fig. 2. This is in contrast to the channel shown on the right, where the Kendall-τ score is close to zero, indicating a full re-ordering of the activations. Correspondingly, the CLIP-δ from initial to final is also much higher, which matches with a visual inspection.

Overall, we notice a substantial correspondence between our visual intuition and the CLIP-δ and Kendall-τ scores. Channels with low scores Kendall-τ and high CLIP-δ tend to change substantially. As illustrated in further examples in Figs 8 and 17, one observed difference in these two metrics is that channels maintaining similar classes in the top images will tend to have a lower CLIP-δ.

Effect of Depth. We now consider how the attack is affected by depth, with results for different layers of AlexNet shown in Table 1. We observe that modifications of the earliest layers are significantly harder to achieve than for later layers as confirmed by the metrics and visual examination. The observed CLIP-δ scores, as well as visual observation, shown in Appx. D, both indicate lower layers' channels are more resilient to this sort of attack.

***Push-Up* Decoy Attack.** We study a more targeted attack objective, namely one that actively pushes a set of selected images into the top activating images for every channel. This is achieved with the loss defined in section 2.1. The loss is non-zero when there exist images outside the set of selected images that activate higher than the selected images we intend to push up.

This targeted attack is likely more challenging than the *push-down* attack, which does not specify what images the top-k should be replaced with. Indeed, the *push-up attack*, if successful, can assign the same interpretation to every channel in a layer, making any interpretation attempt based on top-k images minimally informative.

Fig. 1 shows the result of the *push-up* attack using a collection of images with the ImageNet label "Goldfish" as the decoy set. Further, in Fig. 3, we show additional channels, where the top-5 contain a few or consist entirely of Goldfish images. The metrics in Table 1 also demonstrate substantial change and a low likelihood of *whack-a-mole* behavior. Examining the figure closely, we observe that not only Goldfish, but also images sharing traits with Goldfish images are also boosted, suggesting a degree of generality in the newly imposed selectivity, further explored in Appx. F.

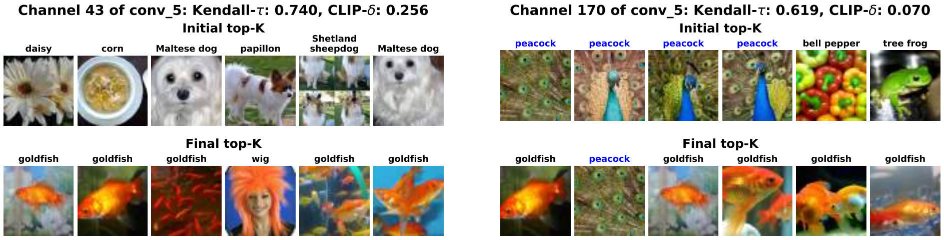


Figure 3: Examples of channels in all-channel *push-up* attack. The decoy images were successfully put in the top images. The Kendall-τ remains relatively high (> 0.5) suggesting much of the channel behavior is preserved while the top activating images completely obfuscate the behavior.

4 Conclusions, Limitations, and Broader Impact

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

References

[1] Pietro Barbiero et al. “Entropy-based logic explanations of neural networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 6. 2022, pp. 6046–6054.

[2] Jasmijn Bastings et al. “"Will You Find These Shortcuts?" A Protocol for Evaluating the Faithfulness of Input Salience Methods for Text Classification”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. 2022, pp. 976–991.

[3] Nick Cammarata et al. “Curve circuits”. In: *Distill* 6.1 (2021), e00024–006.

[4] Nick Cammarata et al. “Curve detectors”. In: *Distill* 5.6 (2020), e00024–003.

[5] Zhi Chen, Yijie Bei, and Cynthia Rudin. “Concept whitening for interpretable image recognition”. In: *Nature Machine Intelligence* 2.12 (2020), pp. 772–782.

[6] MohammadReza Davari et al. “Reliability of CKA as a Similarity Measure in Deep Learning”. In: *The Eleventh International Conference on Learning Representations*. 2022.

[7] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[8] Ann-Kathrin Dombrowski et al. “Explanations can be manipulated and geometry is to blame”. In: *Advances in neural information processing systems* 32 (2019).

[9] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2020.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572* (2014).

[11] Riccardo Guidotti. “Counterfactual explanations and how to find them: literature review and benchmarking”. In: *Data Mining and Knowledge Discovery* (2022), pp. 1–55.

[12] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[13] Juyeon Heo, Sunghwan Joo, and Taesup Moon. “Fooling neural network interpretations via adversarial model manipulation”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[14] Evan Hernandez et al. “Natural Language Descriptions of Deep Visual Features”. In: *International Conference on Learning Representations*. 2022.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. cite arxiv:1503.02531Comment: NIPS 2014 Deep Learning Workshop. 2015. URL: http://arxiv.org/abs/1503.02531.

[16] Jared Kaplan et al. “Scaling laws for neural language models”. In: *arXiv preprint arXiv:2001.08361* (2020).

[17] Been Kim et al. “Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)”. In: *International conference on machine learning*. PMLR. 2018, pp. 2668–2677.

[18] Pang Wei Koh et al. “Concept bottleneck models”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5338–5348.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems* 25 (2012).

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[21] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30 (2017).

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

[23] Neel Nanda et al. “Progress measures for grokking via mechanistic interpretability”. In: *arXiv preprint arXiv:2301.05217* (2023).

[24] Tuomas Oikarinen and Tsui-Wei Weng. “CLIP-Dissect: Automatic Description of Neuron Representations in Deep Vision Networks”. In: *arXiv preprint arXiv:2204.10965* (2022).

[25] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. “Feature Visualization”. In: *Distill* (2017). https://distill.pub/2017/feature-visualization. DOI: 10.23915/distill.00007.

[26] Chris Olah et al. "Zoom In: An Introduction to Circuits". In: *Distill* (2020). https://distill.pub/2020/circuits/zoom-in. DOI: 10.23915/distill.00024.001.

[27] Jayneel Parekh, Pavlo Mozharovskyi, and Florence d'Alché-Buc. "A framework to learn with interpretation". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 24273–24285.

[28] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[29] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[30] Tilman Räukur et al. "Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks". In: *arXiv e-prints* (2022), arXiv–2207.

[31] Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[32] Wen Shen et al. "Interpretable Compositional Convolutional Neural Networks". In: *Proceedings of the International Joint Conference on Artificial Intelligence*. 2021.

[33] Dylan Slack et al. "Fooling lime and shap: Adversarial attacks on post hoc explanation methods". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 180–186.

[34] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[35] Rui Wang, Xiaoqian Wang, and David Inouye. "Shapley Explanation Networks". In: *International Conference on Learning Representations*. 2021.

[36] Jason Yosinski et al. "Understanding neural networks through deep visualization". In: *arXiv preprint arXiv:1506.06579* (2015).

[37] BIN YU. "Stability". In: *Bernoulli* (2013), pp. 1484–1500.

[38] Mert Yuksekgonul, Maggie Wang, and James Zou. "Post-hoc concept bottleneck models". In: *International Conference on Learning Representations*. 2023.

[39] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[40] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. "Interpretable convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8827–8836.

[41] Bolei Zhou et al. "Interpretable basis decomposition for visual explanation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 119–134.

[42] Roland S Zimmermann et al. "How Well do Feature Visualizations Support Causal Understanding of CNN Activations?" In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 11730–11744.

Appendix of the paper entitled “Adversarial Attacks on the Interpretation of Neuron Activation Maximization”

A Related Work

A growing body of literature has investigated the interpretability of Convolutional Neural Networks (CNNs) and their lack of robustness under different manipulations of interpretability methods.
Interpretability methods. Previous work aiming to provide interpretability of DNNs can be grouped into two broad categories. Firstly, some works develop *interpretable-by-design* methods that provide interpretations without relying on external tools. These methods usually couple traditional layers with various types of interpretable components. Examples range from concept explanations [5, 18, 1, 38], feature attributions [35, 27]to part of object disentanglement [40, 32]. Secondly, there are methods usually called *post-hoc* that aim to explain and understand either specific components (e.g., weights, neurons, layers) or outputs of a *trained* DNN. To interpret the output of models for a particular data instance (local interpretability), while feature attribution methods [21, 31] such as saliency maps assign a weight to each input feature corresponding to its importance on the model’s output, counterfactual examples aim to give the minimal changes required to change the model’s output [11]. There are post-hoc approaches that aim to interpret the internal logic of particular DNNs through their components and representations. For example, some methods focus on layer representations through *concept vectors* [17, 41], on sub-network interpretability through *circuits* [2, 3], and individual neurons via e.g., feature visualization. Our work focuses on feature visualization, which is one of the most popular techniques to understand the learned features of individual neurons [42, 25].

Interpretability manipulation. There is a recent trend to analyze the reliability of interpretable techniques through the lens of *stability*. Stability aims to study to what extent the interpretability technique is statistically robust to reasonable input perturbations and model perturbations [13, 37]. Most works that study input and model manipulability focus on feature attributions. For example, [8] design adversarial input perturbations to change feature attributions in a targeted way, and [13] shows that such manipulation can be performed through *adversarial model manipulation*, realized by fine-tuning a pre-trained model to change feature attributions while keeping the same accuracy of the original model. Despite sharing similarities with this work thanks to the use of adversarial model manipulation, instead of studying the manipulability of feature attribution methods, we focus on neuron interpretability, which brings different challenges such as the *whack-a-mole* problem explained in Sec. 2.1.

B Hyperparameters and Training Details

This section presents the details of the hyperparamters and training settings used to run our attacks.

B.1 Push-Up and Push-Down Attacks

We train for 2 epochs over the ImageNet-1k training set with a batch size of 256. We use the *Adam* optimizer with learning rate 1e-5.

Regarding α, we employ a dynamic updating rule inspired by *Algorithm 1: Dynamical balancing of Distillation and CKA map loss* in appendix A of [6] in order to have better control over loss in accuracy. We initialize α as 0.1 (except for on the push-down attack for *conv-2* where using $\alpha = 0.01$ had more stable results). If the accuracy loss is greater than 0.5% we halve the current α. If it is less than 0.1% we double α. With this dynamic update, we aim to minimize the loss in accuracy while still ensuring the top images shifts.

B.2 Optimization Curves

We show in Figure 4 the evolution of attack and maintain losses across two epochs. It can be observed that the attack loss of late layers (conv 4, conv 5) decreases very quickly, and almost monotonically, showing the easiness to attack late layers. In contrast, early layers do not have the same behavior. We can also observe from the training curves that the maintain loss is almost close to its initial value after 2 epochs. This corroborates the observed accuracy preservation as shown in Table 1 of the paper.

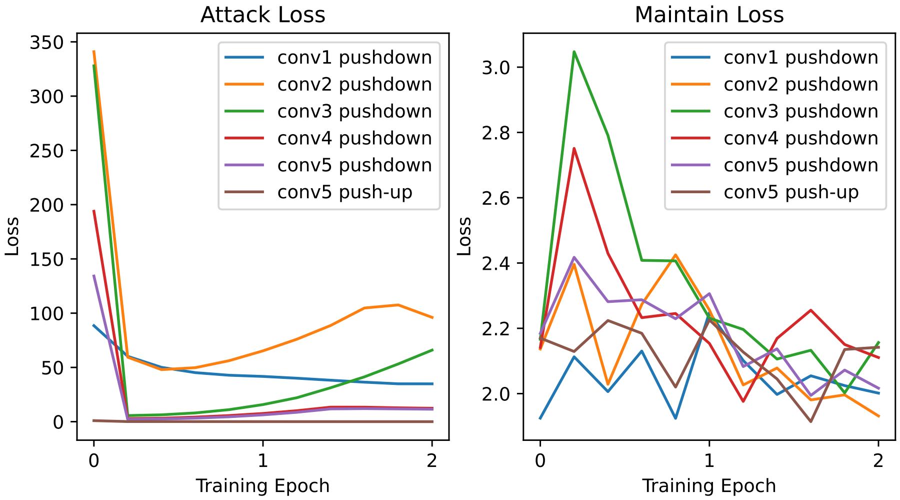


Figure 4: Sample training curves for the maintain and attack objectives. Late layers (conv5, conv4) are easier to attack compared to early ones (conv1, conv2 and conv3). The maintain loss is very close to its initial value after two epochs.

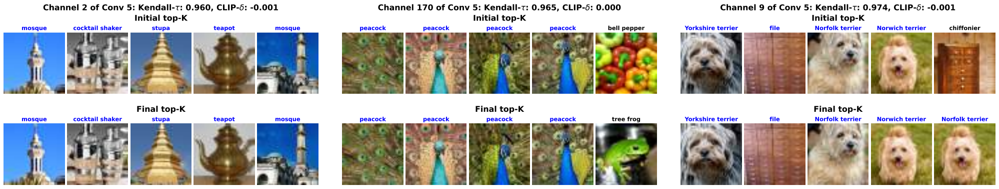


Figure 5: Finetuning baseline result on *Conv5* of AlexNet. All initial images are almost the same after finetuning. Kendall-τ and CLIP-δ are respectively close to 1 and 0, suggesting almost zero changes in channel behavior and semantic changes.

C Additional Results for Push-down Attack on a Single Channel and on all Channels

Before showing additional results for the push-down attacks on a single channel and on all channels simultaneously, we present below the finetnuing baselines.

C.1 Finetuning Baseline

One can observe that the finetuning baseline fails to change top-k images, in particular, top-4 images are exactly the same in the visualized channels. This is also materialized empirically in the kendall-τ coefficients approaching 1, indicating that the ordering of the images has not significantly shifted. The CLIP-δ scores also indicate that the semantic change in the top-k images is minimal. Overall, the finetuning results in an AlexNet that is extremely similar to PyTorch's default with respect to performance and Interpretation via Feature Visualization.

C.2 Push-Down Attack on Single Channel

Figure 7 shows the results of initial top-k images and final ones after running the push-down attack on every single channel. Except for channels 6 and 4 with relatively low CLIP-δ scores, it can be observed that all other channels have semantically different final top-k images compared to the initial ones. This can be also seen by higher values of CLIP-δ scores.

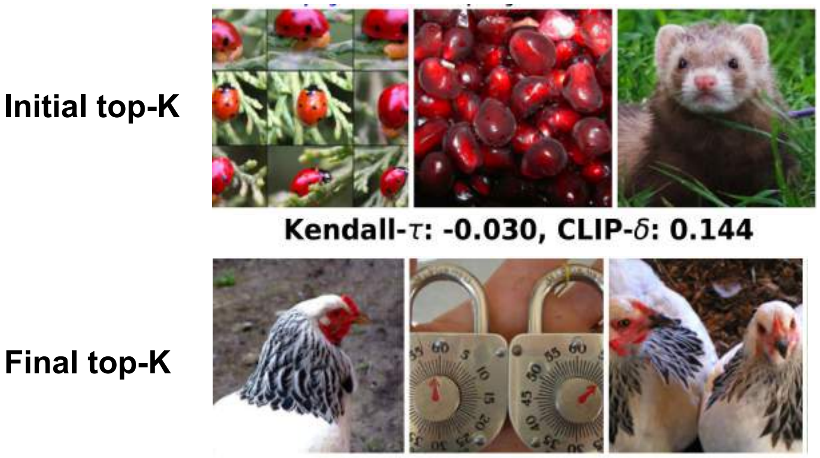


Figure 6: Top images for a channel before and after a single-channel Push-Down attack.

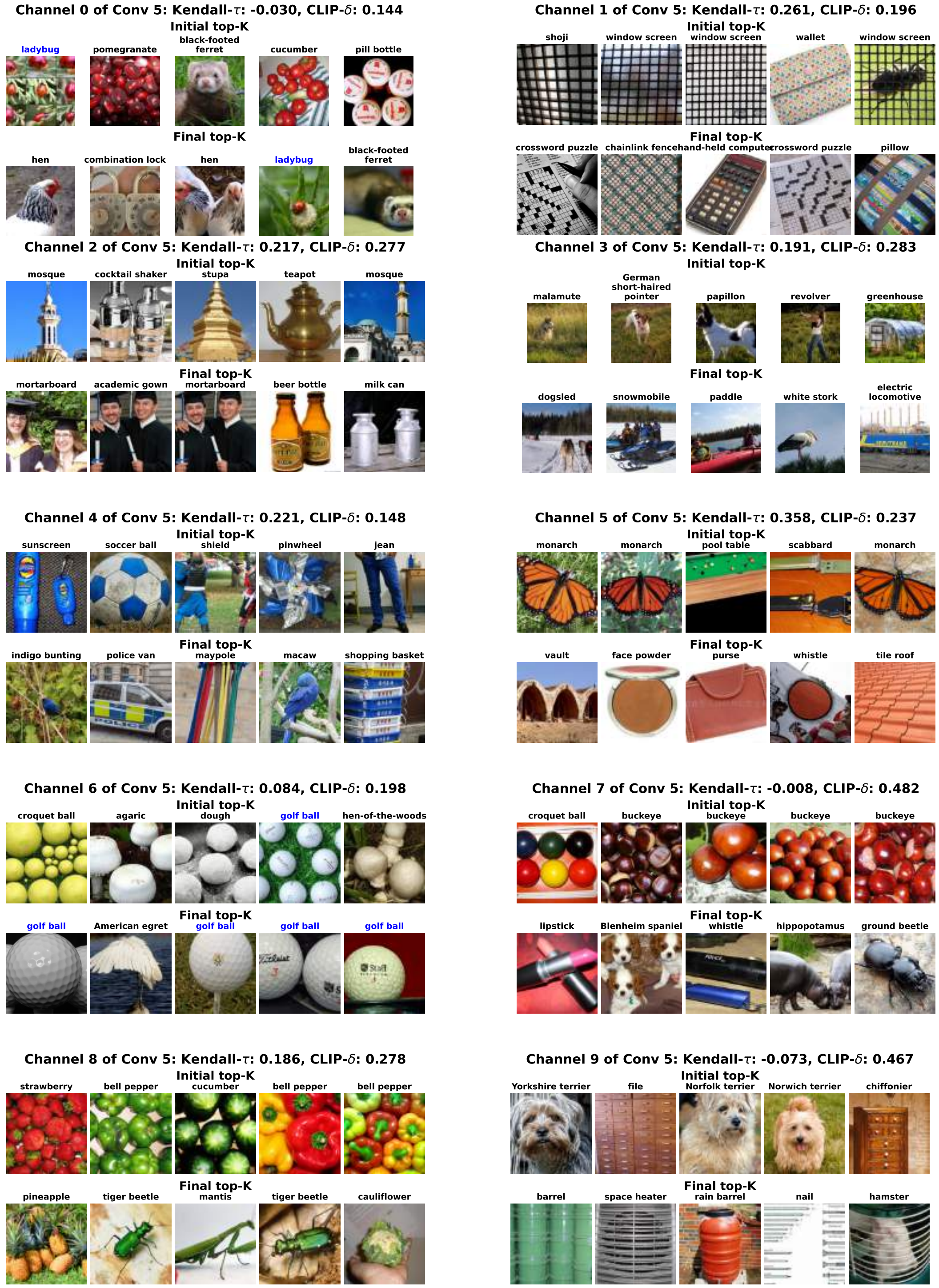


Figure 7: Push-down attack on a single-channel of *Conv5* of AlexNet. All initial images have been replaced by other images.

C.3 Push-down All-Channel Attack

This section presents additional results for the push-down attack on all channels at once. The results are obtained by attacking all the channels of the conv5 layer of AlexNet. We first show visual examples of results obtained from the training set of ImageNet and show its generalization to the validation set.

Visual Examples. Figure 8 shows results obtained on 10 randomly chosen channels. It can be observed that all initial top-5 images were completely removed from the set of top-activating images. Additionally, channels with high CLIP-δ scores such as channels 102 and 132, present semantically different images (initial vs final) with no overlap classes. In contrast, we observe that channels with low CLIP-δ scores such as channels 254 and 227 usually share similar classes in top-activating images. Finally, from Kendall-τ scores, we observe that channels that have high Kendall-τ (e.g., channel 108 and 185) do not often have high values of CLIP-δ scores, indicating that the weak change in channel behavior assessed by the Kendall-τ is often related to low semantic change.

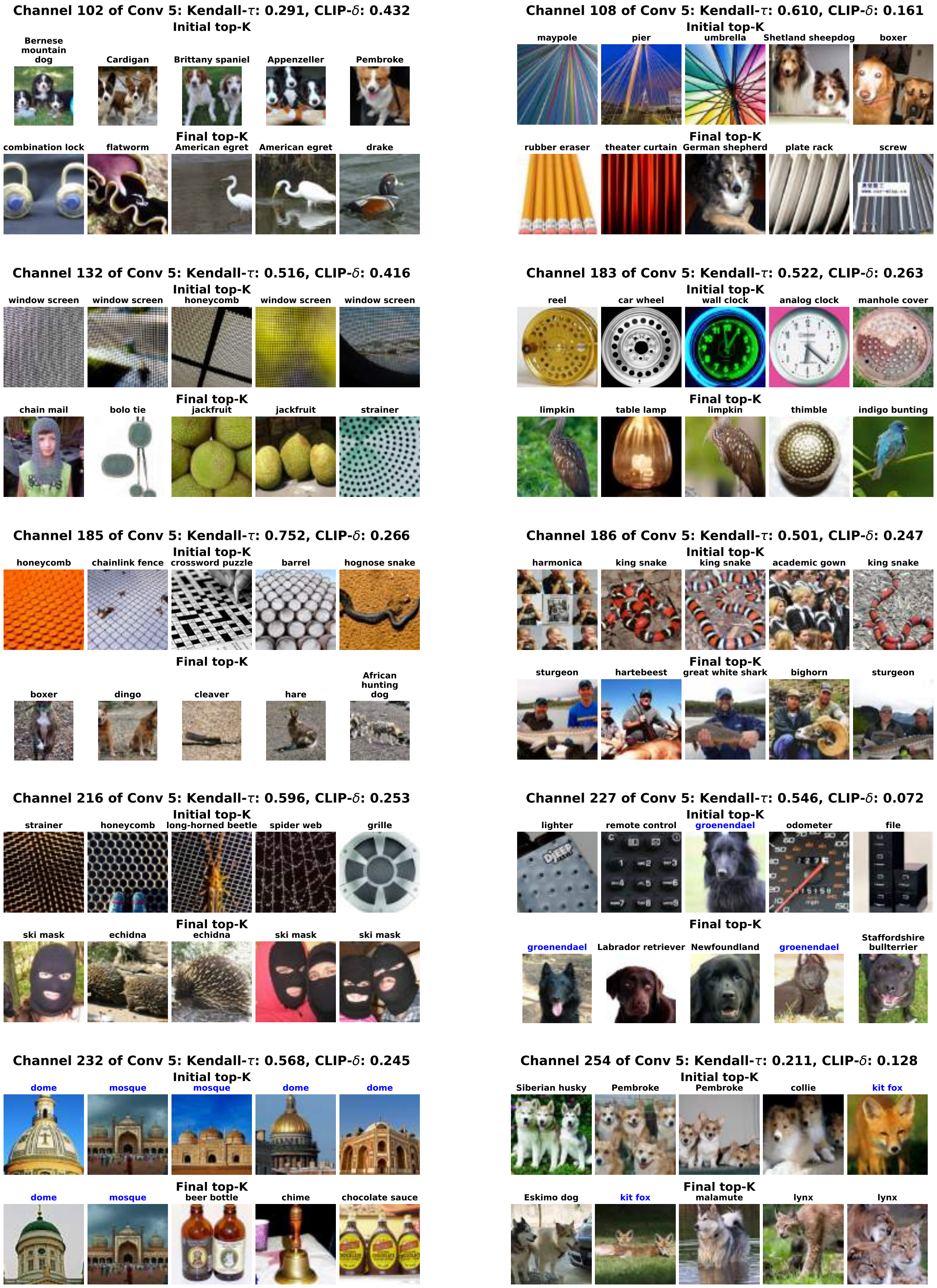


Figure 8: Push-down all-channel attack of *Conv5* of AlexNet. All initial top-5 images were completely removed from the new set of top-5 images, demonstrating the success of the attack. Channel indexes were chosen randomly.

Generalization on Validation Set. We evaluate the generalization of our attack on the validation set of ImageNet. This gives more insights to the change of feature visualization. Figures 9 and 10 show the initial top-k images and final ones from training and validation sets for 10 randomly chosen channels.

It can be observed that on every channel, from the validation set, at least one image from the initial top-5 images is no longer present in final top-5 images (for the majority of these channels, the first top-activating is no longer the top one). We also observe a complete replacement of top-5 images on the validation set when Kenall-τ scores and CLIP-δ are respectively low and high simultaneously (e.g., channels 37 and 50 of Figure 9). Moreover, the general trends in training and validation are similar suggesting the attack is not just memorizing specific images but leading to a generalized change.

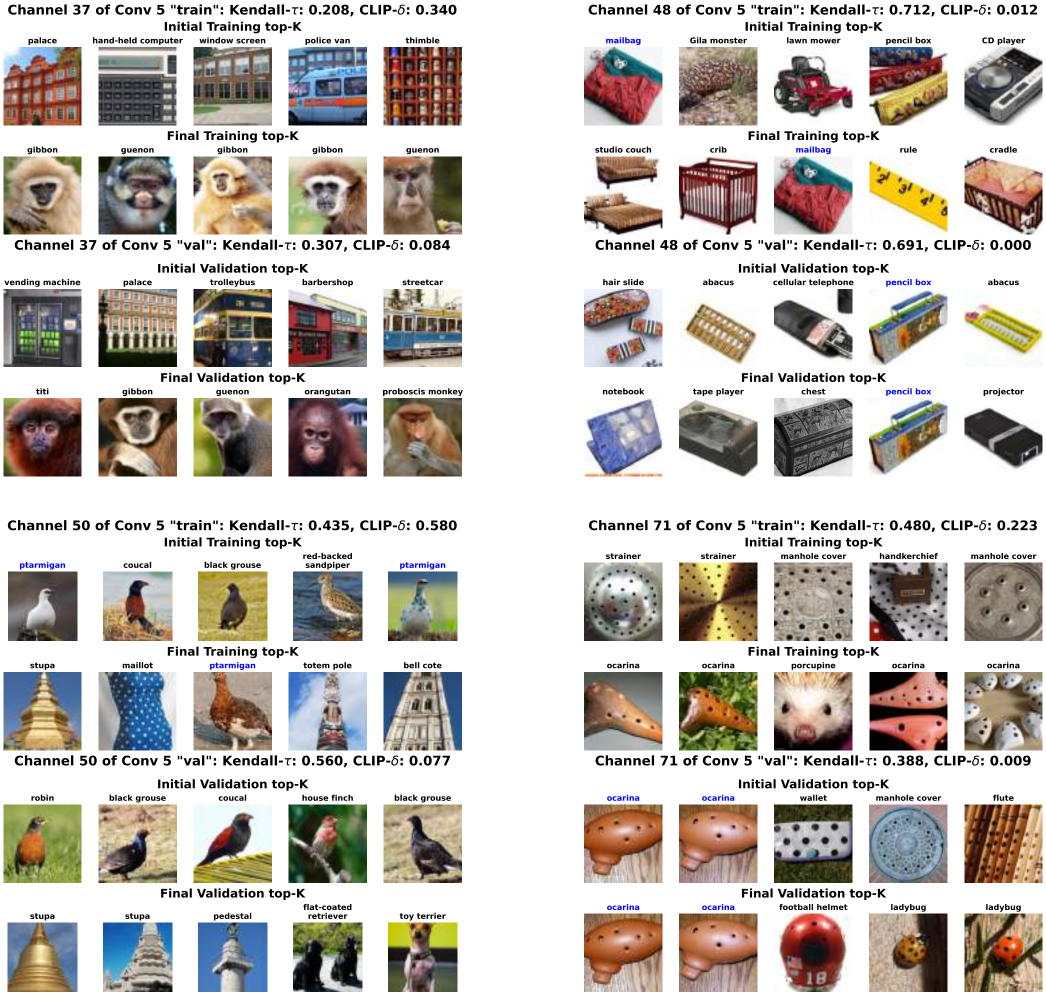


Figure 9: Push-down all-channel attack of *Conv5* of AlexNet. For each channel, the first two rows are top-k images derived from the training set while the last two are derived from the validation set.

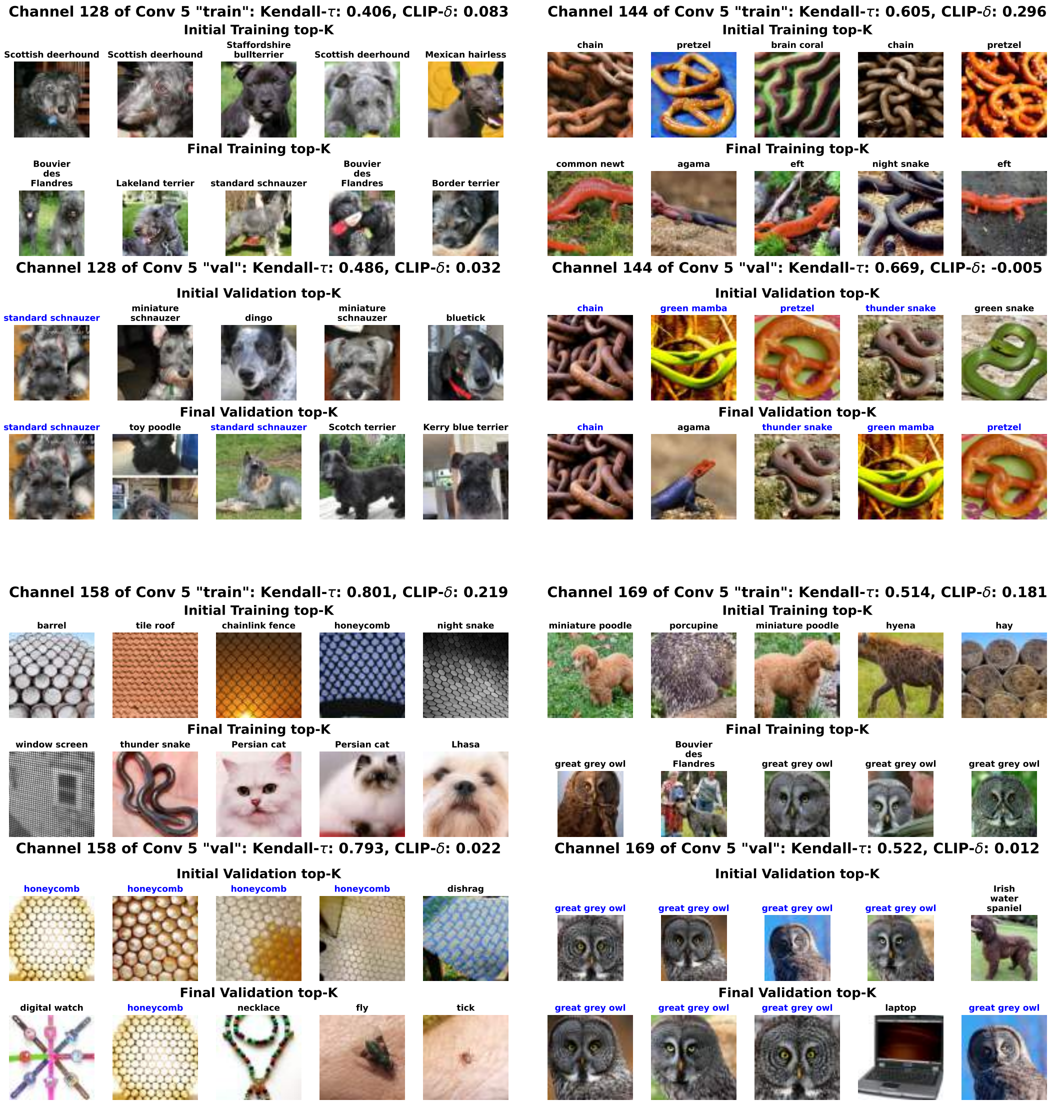
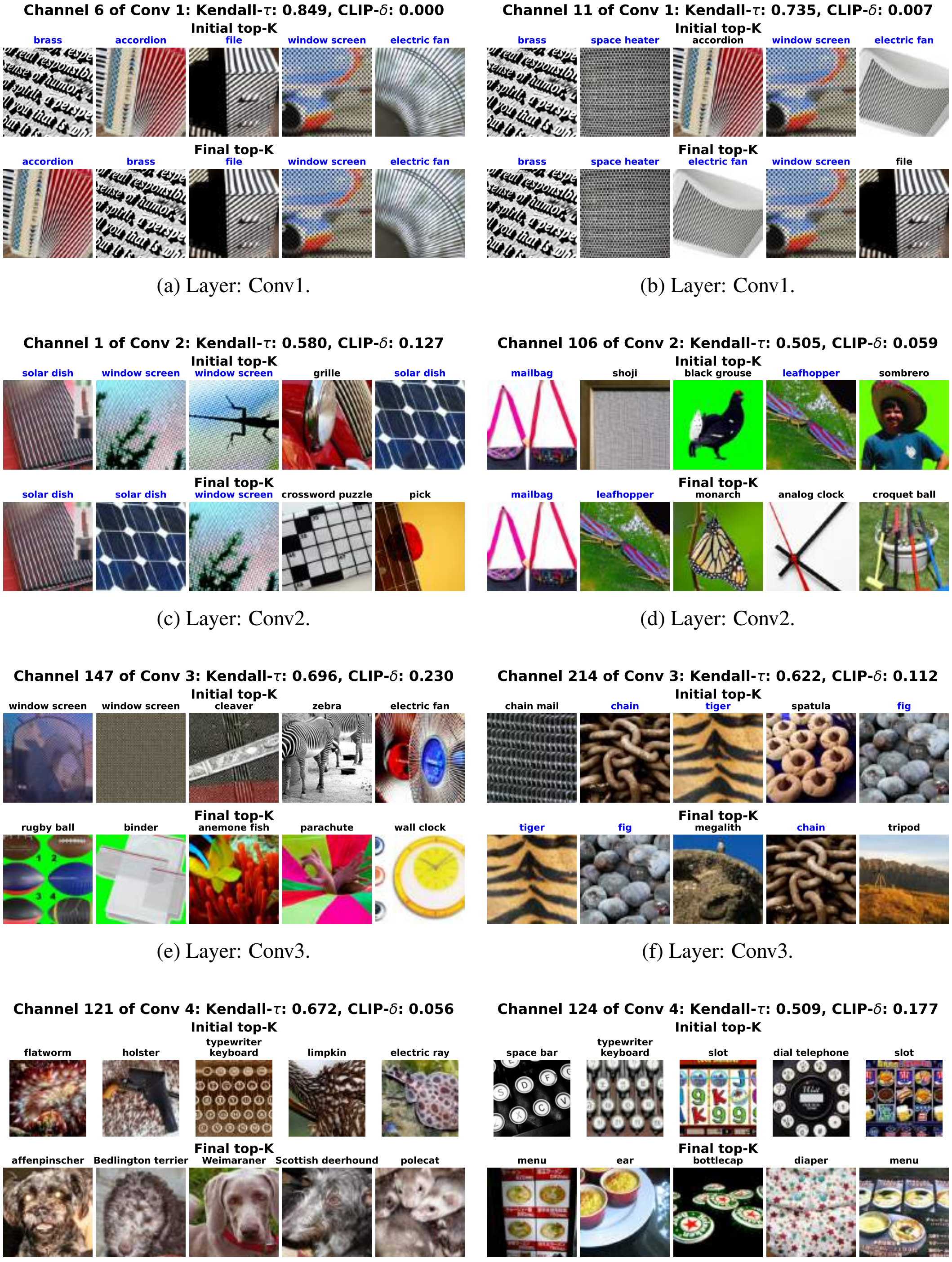


Figure 10: Push-down all-channel attack of *Conv5* of AlexNet. For each channel, the first two rows are top-k images derived from the training set while the last two are derived from the validation set.

D Effect of Depth

We vary different layers of AlexNet and evaluate how the attack is affected by depth. Figure 11 shows results obtained on randomly chosen channels from conv1, conv2, conv3, and conv4 of AlexNet. It can be observed that the earliest layers conv1 and conv2 are harder to attack. This is materialized by high values of Kendal-τ and low values of CLIP-δ scores. When increasing the depth (conv3 and conv4) we observe a complete replacement in top-5 images in channels 147 (conv3), 121 (conv4) and 124 (conv4), although some of these channels have low values of CLIP-δ scores.



(a) Layer: Conv1.

(b) Layer: Conv1.

(c) Layer: Conv2.

(d) Layer: Conv2.

(e) Layer: Conv3.

(f) Layer: Conv3.

(g) Layer: Conv4.

(h) Layer: Conv4.

Figure 11: Push-down all-channel attack of on several layers of AlexNet. Channels indexes were selected randomly. While there are some changes in top-activating images of early layers (conv1 and conv2), they are not significant as materialized by low values of CLIP-δ and high values of Kendall-τ. For conv3 and conv4, we see a complete replacement of top-5 images on channels 147 (conv3), 121 (conv4), and 124 (conv4).

E Whack-a-mole

This section provides further investigations into the existence of the whack-a-mole problem for the push-down attack on AlexNet.

E.1 Further Details for Whack-a-Mole Metrics

We introduce two metrics based on Kendall-τ and CLIP similarity, denoted by Kendall-τ-W_j and CLIP-W_j, where “W” refers to whack-a-mole and j refers to the channel index for which the whack-a-mole effect is assessed, which we naturally compare to all other channels in a layer.

Kendall-τ-W$_j$. Given a channel index j, and using the subset of ImageNet $\mathcal{D}_\tau$, we obtain the maximum Kendall-τ score between rankings $R_{\text{init},j}$ and $R_{\text{final},i}$ where $i \neq j$. To obtain Kendall-τ-W_j, we divide this maximum value by the initial maximum Kendall-τ score i.e. the score over $R_{\text{init},j}$ and $R_{\text{init},i}$ where $i \neq j$. Effectively, we find the most similar other post-attack channel, then normalize against thee most similar pre-attack channel.

CLIP-W$_j$. Using the top-k images in the initial model and channel j we obtain CLIP-$W_j = \max_{i \neq j} \bar{C}_{j,i}^{\text{initial,final}} / \max_{i \neq j} \bar{C}_{j,i}^{\text{initial,initial}}$ comparing to all top-k images in other channels of the final model, normalized against that same similarity metric in the initial CLIP scores.

E.2 Whack-a-mole Results

We further analyze the existence of the whack-a-mole problem by observing Fig. 13 which shows for a given channel of AlexNet *Conv5*, the top-k images in the modified model which have the closest Kendall-τ-W and CLIP-W scores (not including the channel itself).

We observe that the first channel (channel 2 on Fig. 13) has little to no visually discernible similarity to nearby channels in the modified model as well confirmed by the Kendall-τ-W. Indeed a majority of the channels look like this (see Appx.E). On the other hand, we do observe similar images for the initial channel 193 and its nearest final one (163), which was picked as the most illustrative examples ("hard" one) where the red curve of Fig. 14 is above the blue one. However, for this "hard" example, more insight is given by investigating the CLIP-W_j where the denominator notably measures the clip similarity to other channels in the original model. A score of $\lessapprox 1$ suggests that the original model already had a high similarity to another channel. Indeed in the Appx.E, for the second example, we confirm there is a very similar channel in the original model. To gain further insight into CLIP-W_j, in Fig.14, we further visualize the numerator and denominator of CLIP-W_j for all the channels (red line) and sort them by the initial similarity to other channels (denominator). We observe that the red line is usually below the blue line, and if it exceeds, it is not by a large relative amount. This suggests that channels with high whack-a-mole metrics are actually ones that already had similarities to other channels in the original model. Overall we conclude the presence of the whack-a-mole problem is minimal in our current attack.

Zoom onto Channel 193 for Whack-a-mole. We begin by showing the full overview of the behavior of channel 193, selected as one "hard" case where similar initial images are found in final (post-attack) top-k images of another channel. As discussed in Section C.2, although similar initial images for channel 193 were found in channel 163 after the attack, it appears from the second row of Figure 15 that channel 193 was initially highly correlated with the channel 90 according to CLIP-δ score. Moreover, the fact that the CLIP-δ-W_j is $0.991 < 1$ shows that the nearest post-attack channel (channel 163) is not more correlated than the nearest pre-attack channel (channel 90) according to CLIP scores. This, therefore, limits the existence of the whack-a-mole problem on this channel.

Additional Investigation of Potential Existence of Whack-a-mole. These randomly selected examples support the general findings reported in figure-7. While certain channels may have similar top images to specific post-attack channels, it is generally the case that even the most similar channels are distinct. In figure-6 in the main body of text, the two bottom rows denote the top 5 images of the most similar channels to the pre-attack channel measured by the Kendall-τ and CLIP-W_j respectively.

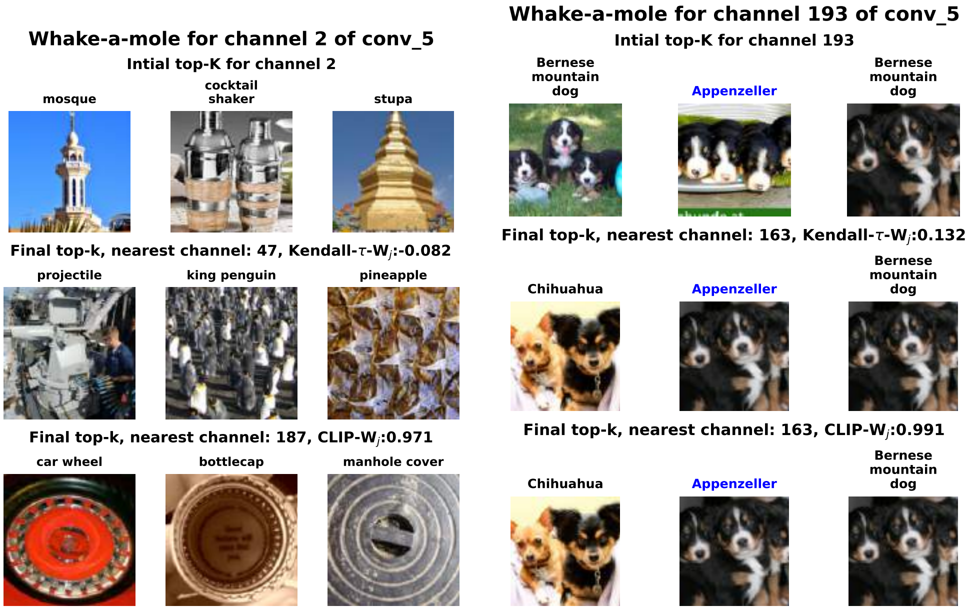

Whake-a-mole for channel 2 of conv_5
Intial top-K for channel 2
mosque
cocktail shaker
stupa
Final top-k, nearest channel: 47, Kendall-τ-W$_j$:-0.082
projectile
king penguin
pineapple
Final top-k, nearest channel: 187, CLIP-W$_j$:0.971
car wheel
bottlecap
manhole cover
Whake-a-mole for channel 193 of conv_5
Intial top-K for channel 193
Bernese mountain dog
Appenzeller
Bernese mountain dog
Final top-k, nearest channel: 163, Kendall-τ-W$_j$:0.132
Chihuahua
Appenzeller
Bernese mountain dog
Final top-k, nearest channel: 163, CLIP-W$_j$:0.991
Chihuahua
Appenzeller
Bernese mountain dog

Figure 13: We show the initial top images for two channels and beneath are the corresponding final top images of closest channels w.r.t Kendall-τ-W$_j$ and CLIP-W$_j$.

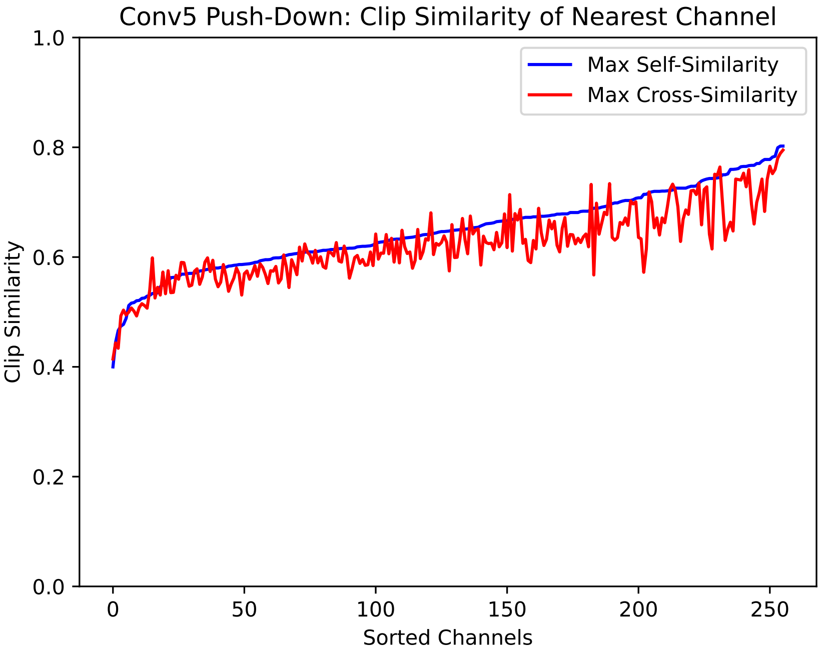


Figure 14: We compare initial CLIP similarity to other channels (blue) versus similarity after attack (red). Red and blue largely track each other for all channels.

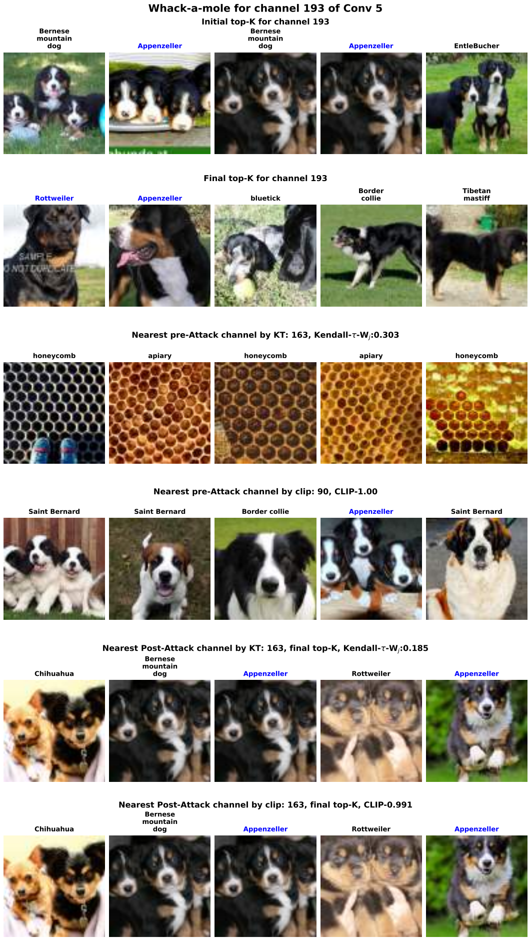
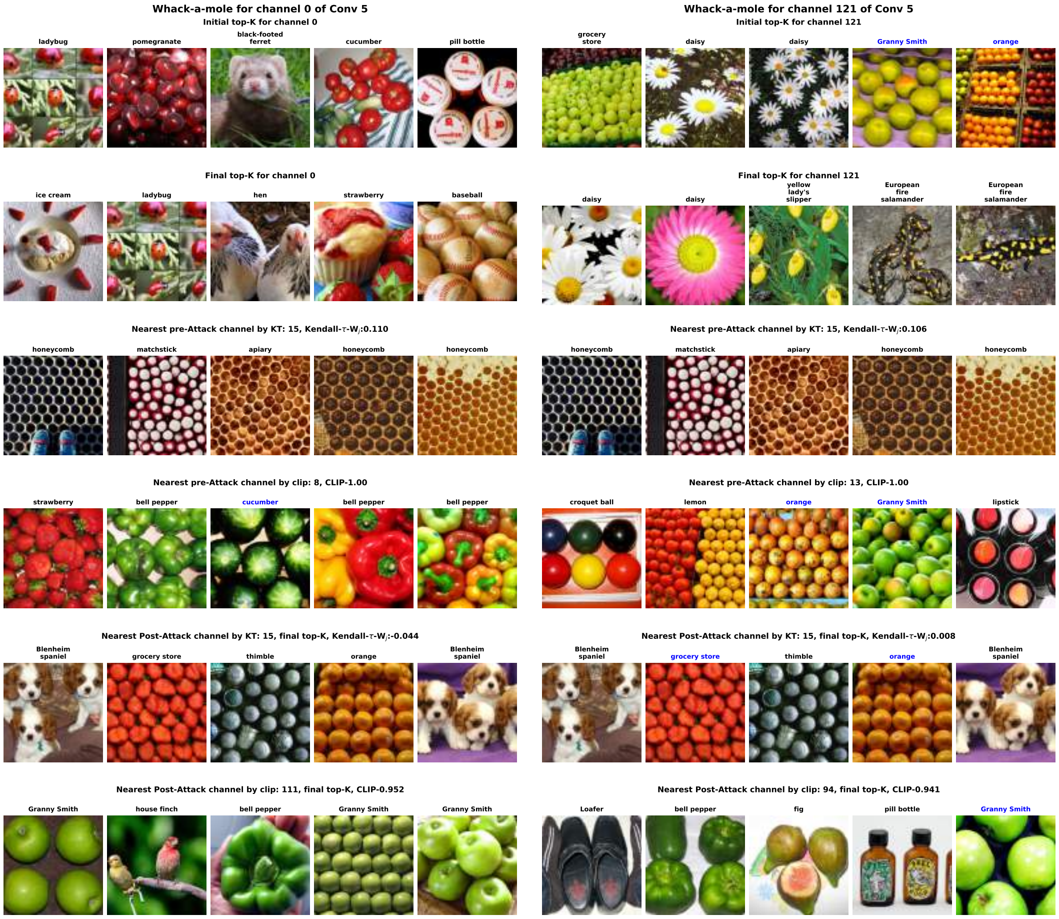


Figure 15: Illustrations for the existence of whack-a-mole on the channel 193, found as one of the high whack-a-mole cases. The first two rows show the initial and final top-k images for the targeted channel. The third and fourth rows show the initial nearest channels w.r.t. Kendall-τ-W_j and CLIP-δ-W_j, respectively. The fifth and sixth rows show the nearest post-attack channel according to Kendall-τ-W_j and CLIP-δ-W_j respectively.


Whack-a-mole for channel 0 of Conv 5
Initial top-K for channel 0
ladybug
pomegranate
black-footed ferret
cucumber
pill bottle
Final top-K for channel 0
ice cream
ladybug
hen
strawberry
baseball
Nearest pre-Attack channel by KT: 15, Kendall-τ-W$_j$:0.110
honeycomb
matchstick
apiary
honeycomb
honeycomb
Nearest pre-Attack channel by clip: 8, CLIP-1.00
strawberry
bell pepper
cucumber
bell pepper
bell pepper
Nearest Post-Attack channel by KT: 15, final top-K, Kendall-τ-W$_j$:-0.044
Blenheim spaniel
grocery store
thimble
orange
Blenheim spaniel
Nearest Post-Attack channel by clip: 111, final top-K, CLIP-0.952
Granny Smith
house finch
bell pepper
Granny Smith
Granny Smith
Whack-a-mole for channel 121 of Conv 5
Initial top-K for channel 121
grocery store
daisy
daisy
Granny Smith
orange
Final top-K for channel 121
daisy
daisy
yellow lady's slipper
European fire salamander
European fire salamander
Nearest pre-Attack channel by KT: 15, Kendall-τ-W$_j$:0.106
honeycomb
matchstick
apiary
honeycomb
honeycomb
Nearest pre-Attack channel by clip: 13, CLIP-1.00
croquet ball
lemon
orange
Granny Smith
lipstick
Nearest Post-Attack channel by KT: 15, final top-K, Kendall-τ-W$_j$:0.008
Blenheim spaniel
grocery store
thimble
orange
Blenheim spaniel
Nearest Post-Attack channel by clip: 94, final top-K, CLIP-0.941
Loafer
bell pepper
fig
pill bottle
Granny Smith

(a) Targeted channel: 0.

(b) Targeted channel: 121.

Figure 16: Illustrations for the existence of whack-a-mole on two randomly chosen channels. The first two rows show the initial and final top-k images for the targeted channel. The third and fourth rows show the initial nearest channels w.r.t. Kendall-τ-W_j and CLIP-W_j, respectively. The fifth and sixth rows show the nearest post-attack channel according to Kendall-τ-W_j and CLIP-W_j, respectively.

F Additional Illustrations for the Push-up Attack

This section provides additional visual illustrations of the push-up all-channel attack on the layer conv5 of AlexNet.

Visual Examples. We first provide additional visual illustrations in Figure 17 of the attack on 10 randomly chosen channels. As a reminder, this push-up attack aims to make images of the Goldfish class appear in the top-k images of every channel on the targeted layer. From Figure 17, a first observation is the fact that out of these 10 randomly chosen channels, only two channels (channel 15 and channel 23) do not show an image with the Goldfish class. On the rest of the channels, an image with Goldfish was successfully inserted in the final top images. Furthermore, in several cases (channels 110, 125, 145, 180, 183, and 50) is the majority class of final top-5 images, demonstrating the success of this attack. It is also important to note the complete replacement of images with the Goldfish class in some channels (e.g., channel 125).

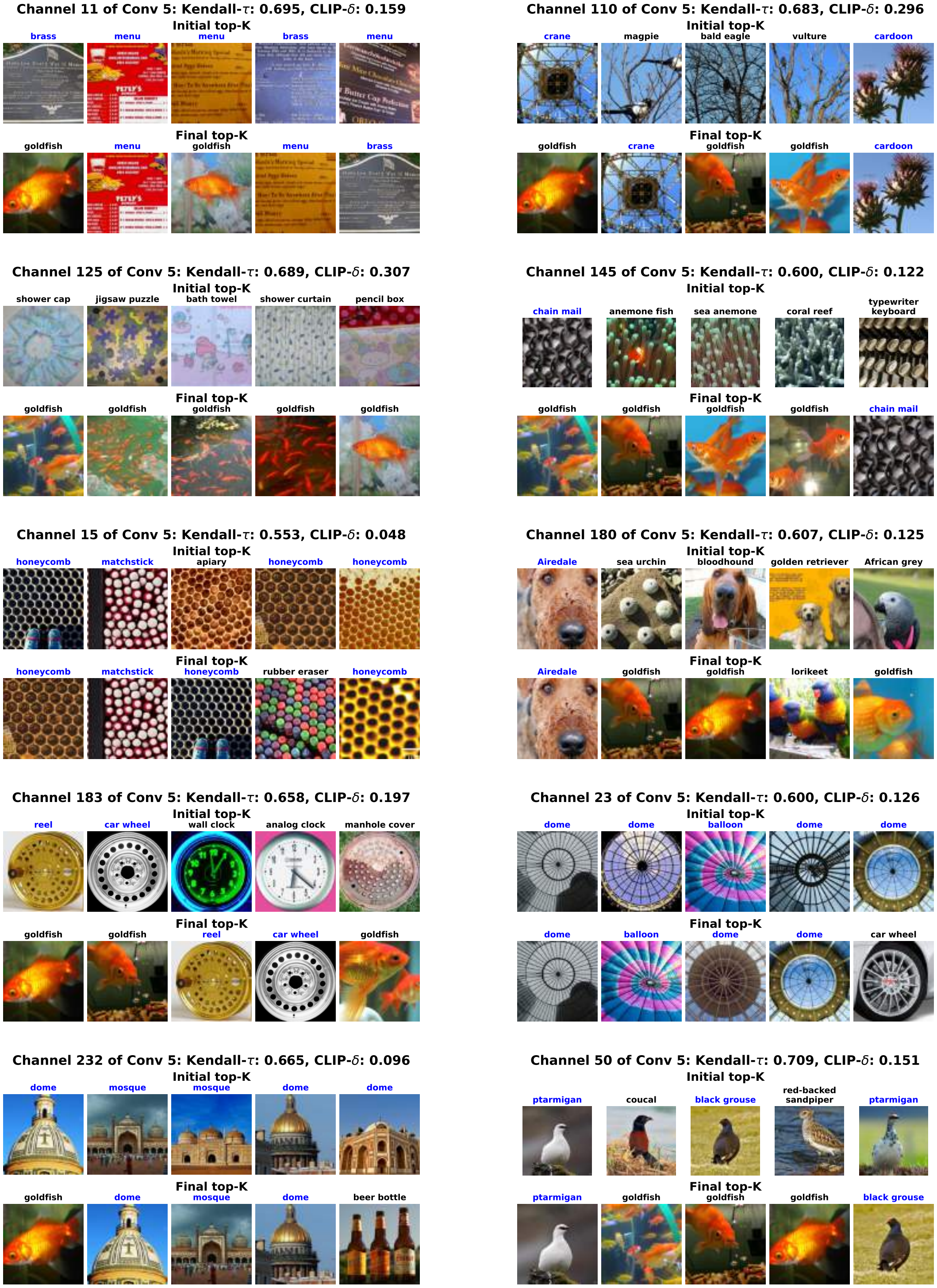


Figure 17: Push-up all-channel attack of *Conv5* of AlexNet. Channel indexes were taken randomly.

Generalization for the Push-Up attack. After demonstrating the success of achieving target manipulability of top-k feature visualization through the push-up attack on training images, it is also important to evaluate whether this success generalizes to unseen data. Figure 18 shows not only top-k images from the training but also from the validation set of ImageNet. We can observe that on all the 10 randomly chosen channels not only at least one image of the Goldfish class is present in the final top-5 images of the training but also at least one image of the Goldfish class is in the final top-5 images from the validation set. Moreover, we also observe a similar number of images of the Goldfish class present in top-5 images from both training and validation sets. This indicates the ability of the push-up attack to generalize on the same distribution from where training examples were drawn.

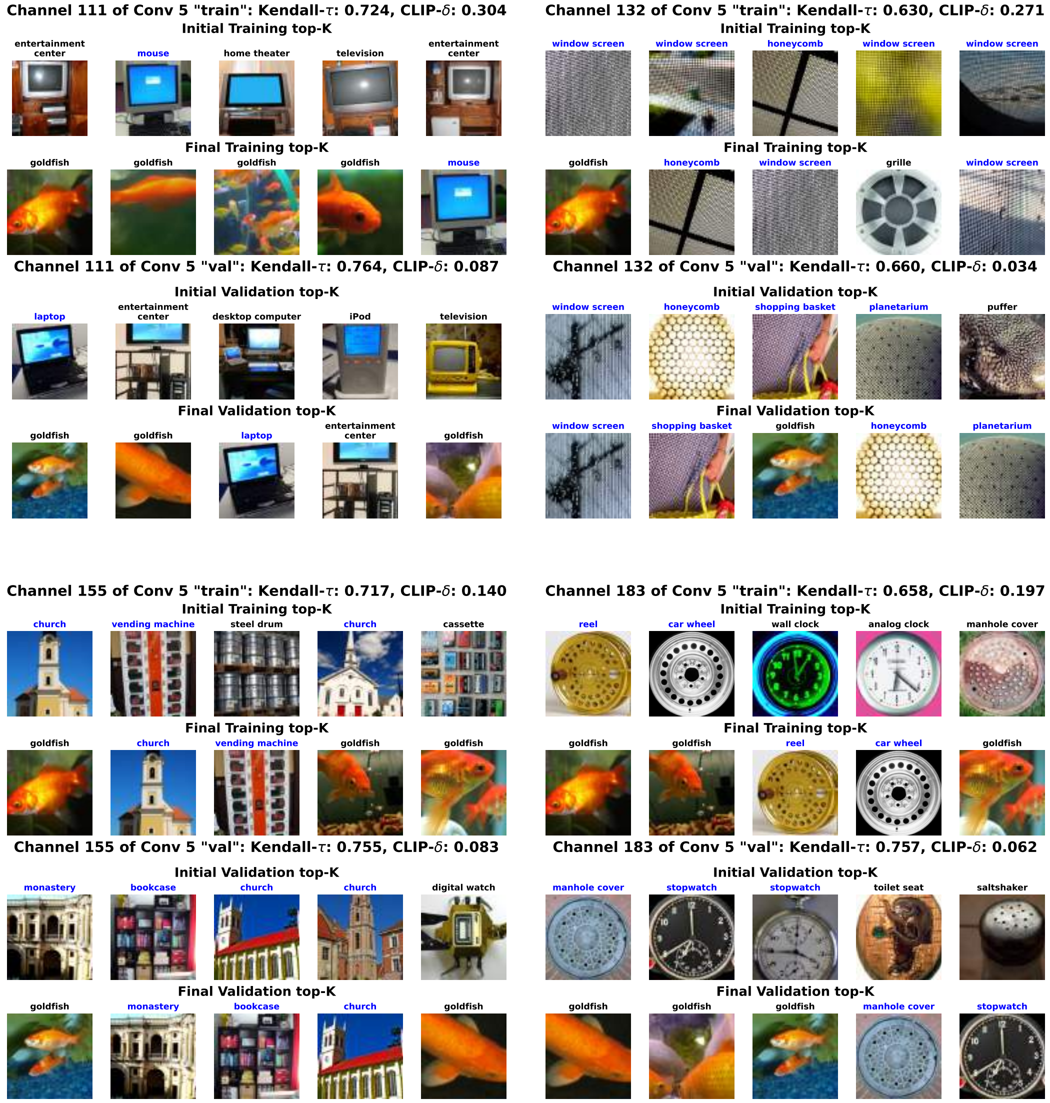


Figure 18: Push-up all-channel attack of Conv5 of AlexNet. For each channel, the first two rows are top-k images derived from the training set while the last two are derived from the validation set.

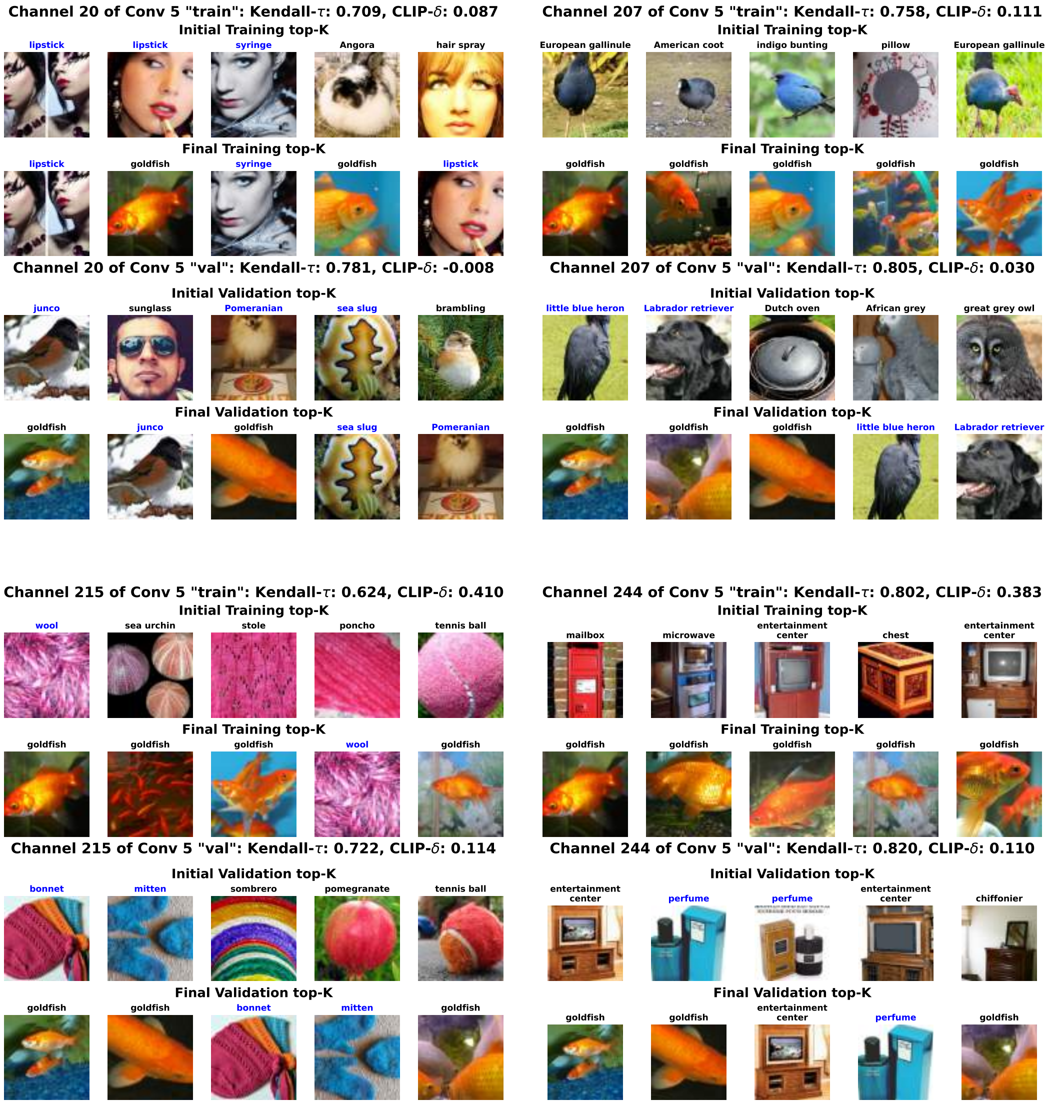


Figure 19: Push-up all-channel attack of Conv5 of AlexNet. For each channel, the first two rows are top-k images derived from the training set while the last two are derived from the validation set.

G Ablation Study on EfficientNet

It is important to show that the proposed attack methodology is not limited to AlexNet. To show that the attack can work on newer, more sophisticated neural nets, we have also run an ablation study on EfficientNet [34]. We select the third convolutional block in the Feature 7 layer and perform a push-down attack similar way to AlexNet. The visual results are shown in Appendix B and the metrics for the layer are given in Table 1 in the main text. We observe similar effects to AlexNet; the top images are changed in terms of the exact images and the semantic concepts. We also observe relatively strong CLIP-δ and Kendall-τ changes. Having confirmed the generality of our approach in this way, we leave a survey study over all relevant architectures to future work, computation power permitting.

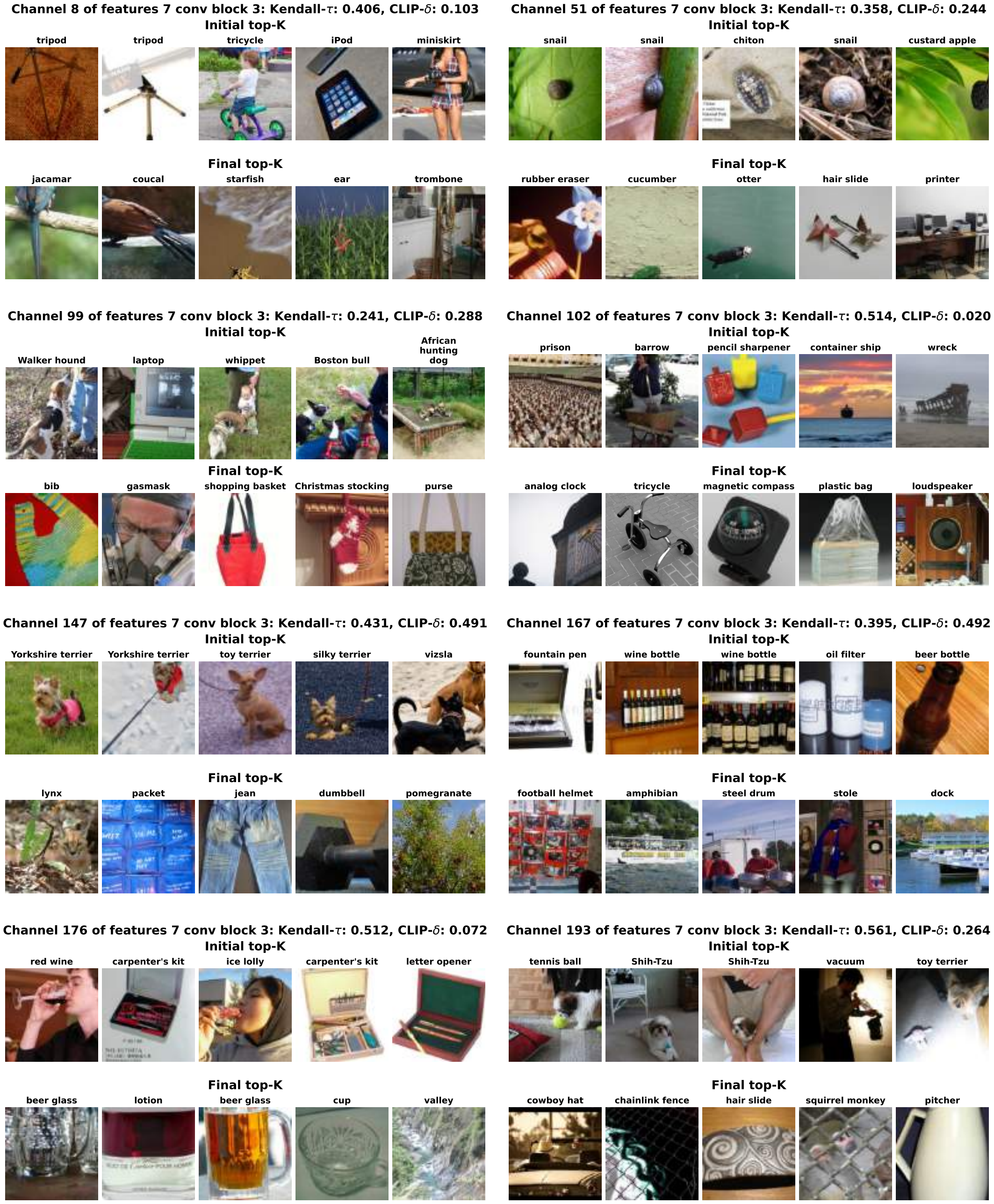


Figure 20: Push-down all-channel attack on Feature 7 block 3 of EfficientNet. All initial top-5 images were completely removed from the new set of top-5 images, demonstrating the success of the attack. Channel indexes were randomly chosen.

H Non-ConvNet Ablations

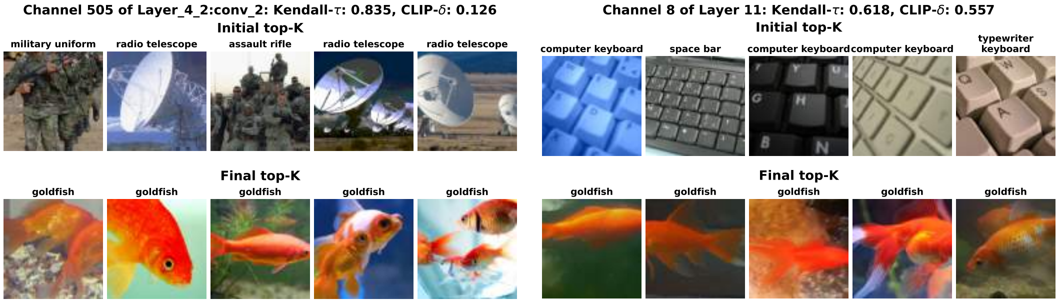
We perform additional ablations on ViT and ResNet-50 to demonstrate the generalizability of our attack framework beyond convolutional neural nets.

H.1 ResNet-50

We have performed additional experiments on Resnet-50 for push-up and push-down attacks, for all the channels of the layer *layer_4_2_conv_2*. We observe no significant loss in accuracy as shown in Table 2. Figures 21 and 22 show the result for a randomly chosen channel. We observed that the results follow similar trends to those for AlexNet, with higher Kendall-τ values on the push up attack, higher CLIP-δ on the pushdown, and low performance loss overall.

H.2 ViT-B/32

We attack the self-attention encoder layer in layers 0, 6, and 11 of ViT-B/32 to present a cross section of the attack behaviour. Visualizations of the results can be seen in figures 21 and 22. Overall the results follow those of AlexNet with later layers having a larger semantic change as shown by the CLIP-δ scores. Interestingly, we note overall increased CLIP-δ scores, in particular for the first layer, whose analogue in AlexNet saw much smaller changes in its feature visualization.



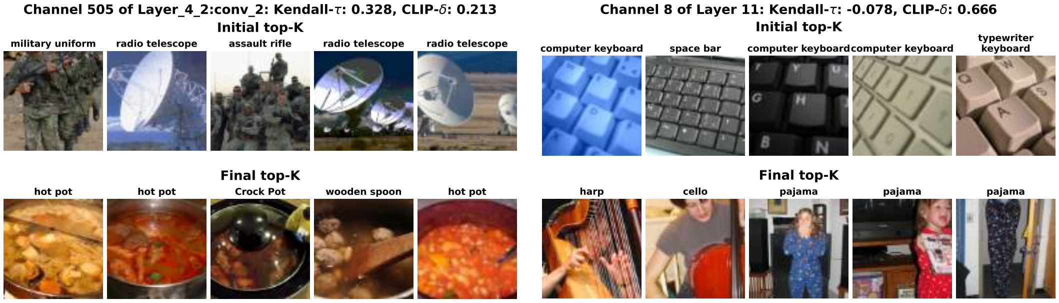
(a) Model: ResNet50

(b) Model: ViT-B/32.

Figure 21: All-channel push-up attacks on ResNet50 and ViT. Goldfish images were successfully put in top images.

Layer/Attack	CLIP-δ	Kendall-τ	CLIP-W	K.-τ-W	Accuracy	Δ Acc.
ViT layer 11 Push Up	0.295	0.399	0.813	0.138	75.7%	-0.22%
ViT layer 11 Push Down	0.378	-0.168	0.833	-0.082	75.6%	-0.27%
ViT layer 6 Push Down	0.244	-0.152	0.885	0.122	75.2%	-0.73%
ViT layer 0 Push Down	0.219	-0.139	0.913	0.133	75.4%	-0.55%
ResNet-50 layer4.2.conv2 Push Down	0.267	0.319	0.946	0.124	80.2%	-0.01%
ResNet-50 layer4.2.conv2 Push Up	0.138	0.784	0.965	0.135	80.2%	-0.01%

Table 2: ViT-B/32 and Resnet-50 with Push-up and Push Down Attacks. Each row reports the result obtained after attacking all units of a particular layer. Note that on ViT, the attacks are quite successful, more than those performed on AlexNet, based on increased clip-δ scores and low accuracy loss. We further note that compared to early AlexNet layers, earlier layers of ViT are less resilient to attacks.



(a) Network: ResNet50.

(b) Network: ViT-B/32.

Figure 22: All-channel push-down attacks on Resnet-50 and ViT. Initial top images were successfully replaced.

Attack	Layer	CLIP-δ	Kendall-τ	CLIP-W	K.-τ-W	Accuracy	ΔAcc
All-Layer Push Down	Conv 1	0.025	0.779	0.981	0.295	56.1%	-0.45%
	Conv 2	0.097	0.447	0.993	0.157		
	Conv 3	0.154	0.512	0.969	0.134		
	Conv 4	0.180	0.558	0.953	0.135		
	Conv 5	0.194	0.584	0.969	0.060		
All-Layer Push Up	Conv 1	0.021	0.726	0.981	0.303	56.1%	-0.46%
	Conv 2	0.049	0.420	0.992	0.137		
	Conv 3	0.070	0.307	0.987	0.108		
	Conv 4	0.170	0.272	0.971	0.097		
	Conv 5	0.248	0.541	0.938	0.068		

Table 3: Alexnet All-Layer Attacks. Each block of rows (for the push-down and push-up attack) shows the results obtained after attacking all the channels and layers of conv layers in AlexNet. We see that both attacks follow the previously seen trend of later layers being easier to attack. Based on a comparison of these metrics against those found in Table 1, we see that the push-down attack is slightly less effective overall, while the push-up attack is actually more effective.

I All-Layer Attack

We perform additional experiments to attack all the channels of every layer simultaneously. Table 3 reports the computed metrics. The Push Down All-Channel Attack has results for each of its layers that correspond well to what we saw in each layers' individual attacks in the main paper (Table 1). Overall the CLIP-δ scores are slightly lower, which is not unexpected as this attack demands a shift in the neurons of all layers leaving less room for compensation than a single layer attack. The Push Up Attack however, actually shows better results in this paradigm. We hypothesize that this is due to synergistic effects in pushing up the same set of images across all layers.

J Synthetic Feature Visualization

J.1 Synthetic Feature Visualization

We study the impact of the push-down and push-up attacks on the synthetic activation-maximizing images of the channels under attack. Synthetic activation-maximizing images are the result of an optimization problem over input pixels solved by gradient ascent on the channel activation under a norm constraint in pixel space. To avoid adversarial noise samples [10] it is necessary to jitter the input image or parameterize it as a smooth function[25].

In Fig. 23, we study the synthetic optimal images for several channels before and after the attack. By visual inspection, while the top-k images change drastically, the synthetic optimal image is largely unaffected. The most common observed change (see also Appx. J) for *Conv5* is a low-frequency modulation of the pattern. We hypothesize that this is because the top-k attack most significantly modifies the weights of the attacked layer, which is a later layer preceded by several downsampling operations.

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

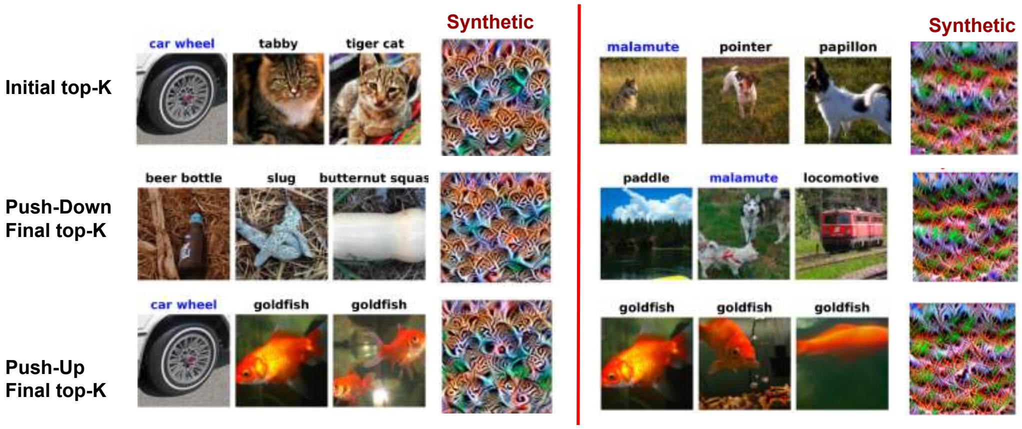


Figure 23: Synthetic feature visualization after attack. We observe the visualization is largely decorrelated to top-k natural images.

J.2 Additional Illustrations for Synthetic Feature Visualization

This section provides illustrations of the decorrelation between synthetic and natural (through top-k images) feature visualization.

Figure 24 shows the natural and synthetic feature visualization before and after the attack on 6 randomly chosen channels of conv5 of AlexNet. We observe a lack of change in the synthetic optimal image, even when top images have been completely replaced by images of the Goldfish class, e.g., in channel 54. We, therefore, reemphasize that attacking the natural feature visualization does not transpose to attacking the synthetic feature visualization. This indicates a decorrelation between the synthetic feature visualization and the top-k images.

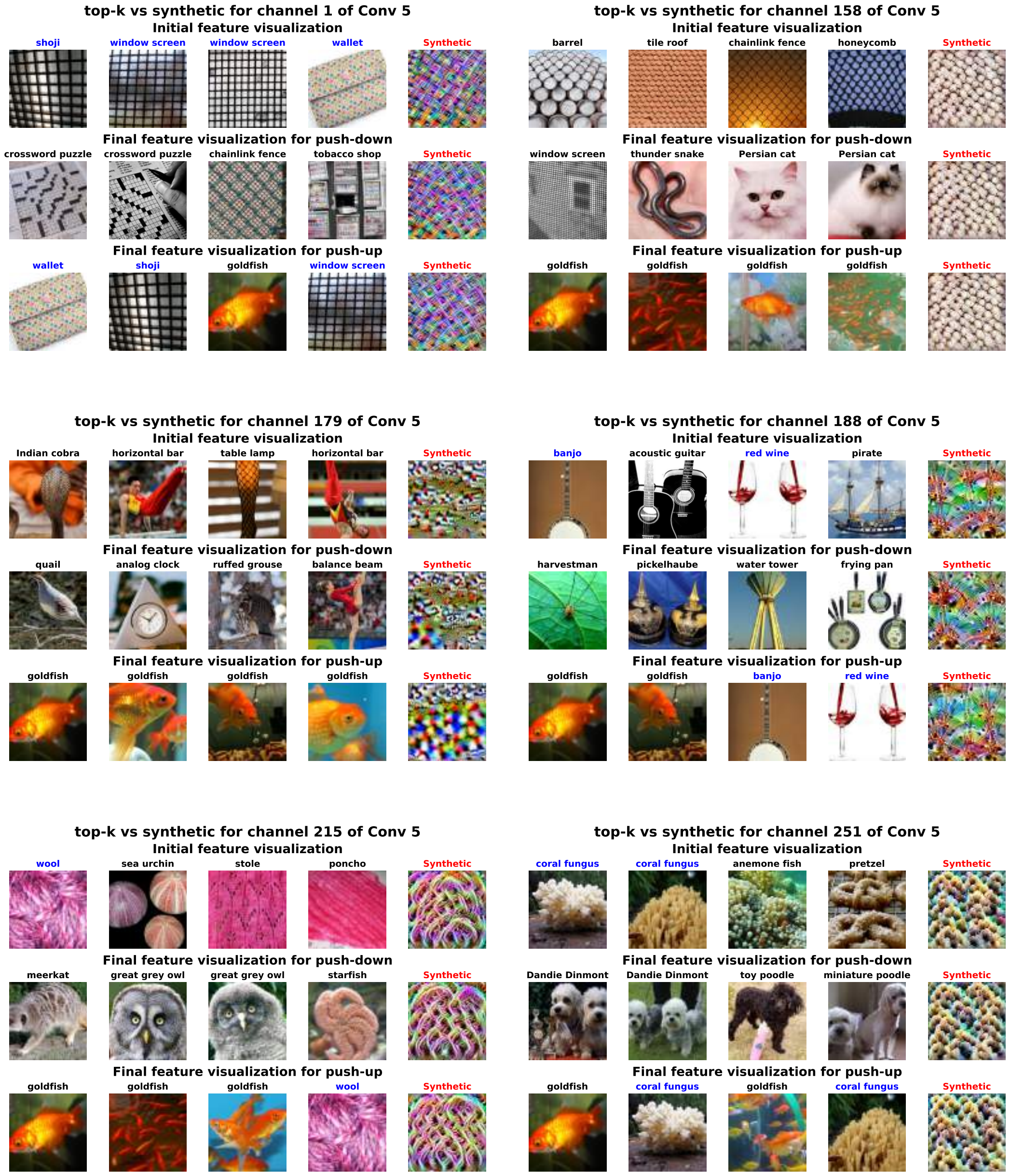

top-k vs synthetic for channel 1 of Conv 5
Initial feature visualization
shoji
window screen
window screen
wallet
Synthetic
Final feature visualization for push-down
crossword puzzle
crossword puzzle
chainlink fence
tobacco shop
Synthetic
Final feature visualization for push-up
wallet
shoji
goldfish
window screen
Synthetic
top-k vs synthetic for channel 158 of Conv 5
Initial feature visualization
barrel
tile roof
chainlink fence
honeycomb
Synthetic
Final feature visualization for push-down
window screen
thunder snake
Persian cat
Persian cat
Synthetic
Final feature visualization for push-up
goldfish
goldfish
goldfish
goldfish
Synthetic
top-k vs synthetic for channel 179 of Conv 5
Initial feature visualization
Indian cobra
horizontal bar
table lamp
horizontal bar
Synthetic
Final feature visualization for push-down
quail
analog clock
ruffed grouse
balance beam
Synthetic
Final feature visualization for push-up
goldfish
goldfish
goldfish
goldfish
Synthetic
top-k vs synthetic for channel 188 of Conv 5
Initial feature visualization
banjo
acoustic guitar
red wine
pirate
Synthetic
Final feature visualization for push-down
harvestman
pickelhaube
water tower
frying pan
Synthetic
Final feature visualization for push-up
goldfish
goldfish
banjo
red wine
Synthetic
top-k vs synthetic for channel 215 of Conv 5
Initial feature visualization
wool
sea urchin
stole
poncho
Synthetic
Final feature visualization for push-down
meerkat
great grey owl
great grey owl
starfish
Synthetic
Final feature visualization for push-up
goldfish
goldfish
goldfish
wool
Synthetic
top-k vs synthetic for channel 251 of Conv 5
Initial feature visualization
coral fungus
coral fungus
anemone fish
pretzel
Synthetic
Final feature visualization for push-down
Dandie Dinmont
Dandie Dinmont
toy poodle
miniature poodle
Synthetic
Final feature visualization for push-up
goldfish
coral fungus
goldfish
coral fungus
Synthetic

Figure 24: Synthetic Feature Visualization attack after push-down and push-up attacks on Conv5 of AlexNet. Channels indexes were taken randomly. We observe a decorrelation between natural top-activating images and synthetic optimal images.