# Interpretation of Neural Networks is Fragile

**Amirata Ghorbani***
Department of Electrical Engineering
Stanford University, CA, USA
amiratag@stanford.edu

**Abubakar Abid***
Department of Electrical Engineering
Stanford University, CA, USA
a12d@stanford.edu

**James Y. Zou**†
Department of Biomedical Data Science
Stanford University, CA, USA
jamesz@stanford.edu

## Abstract

In order for black-box models to be deployed and trusted in many applications, it is crucial to be able to reliably explain why the model makes certain predictions. A fundamental question is: *how much can we trust the interpretation itself?* In this paper, we show that interpretation of deep learning predictions is extremely fragile in the sense that two perceptively indistinguishable inputs with the *same* predicted label can be assigned very *different* interpretations. We systematically characterize the fragility of several widely-used feature-importance and exemplar-based interpretation methods to show that even small random perturbations can affect interpretations, while new systematic perturbations can lead to dramatically different interpretations without changing the label.

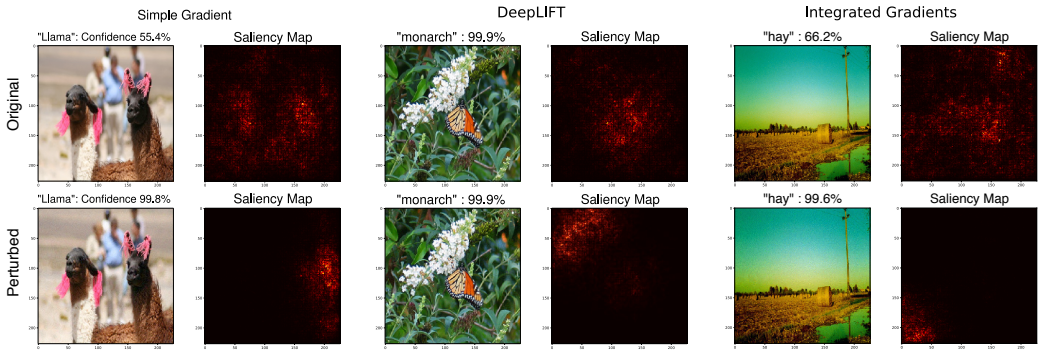
## 1 Introduction

As machine learning algorithms become increasingly complex, explanations for why an algorithm makes certain decisions are ever more crucial. For example, if an AI system predicts a given pathology image to be malignant, then the doctor would want to know what features in the image led the algorithm to this classification. Therefore having interpretations for why certain predictions are made is critical for establishing trust and transparency between the users and the algorithm (Lipton, 2016). The explanation itself, however, must be robust in order to establish human trust. Take the pathology predictor: it would be highly disconcerting if, for a visually indistinguishable image with the same prediction, a very different section is interpreted as being salient. Thus, even if the predictor is robust (both images are correctly labeled as malignant), that the interpretation is fragile would still be highly problematic in deployment.

**Our contributions.** In this paper, we show that widely-used neural network interpretation methods are fragile in the following sense: perceptively indistinguishable images that have the same prediction label by the neural network can often be given substantially different interpretations. We systematically investigate two classes of interpretation methods: methods that assign importance scores to each feature (this includes simple gradient (Simonyan et al., 2013), DeepLift (Shrikumar et al., 2017), and integrated gradient (Sundararajan et al., 2017)), as well as a method that assigns importance scores to each training example: influence functions (Koh & Liang, 2017). For both classes of interpretations, we show that the importance of individual features or training examples is highly fragile to even small random perturbations to the input image. Moreover we show how targeted perturbations can lead to dramatically different global interpretations (Fig. 1).

*Equal contribution †Corresponding author

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.




(a) (b) (c)

Figure 1: **The fragility of feature-importance maps.** Using three popular feature importance methods, **top row** shows the the original images and their saliency maps and the **bottom row** shows the perturbed images (using the center attack with $\epsilon = 8$, as described in Section 3) and the corresponding saliency maps. In all three images, the predicted label has not changed due to perturbation while the saliency maps of the perturbed images are meaningless.

## 2 Interpretation Methods for Neural Network Predictions

### 2.1 Feature-Importance Interpretation

Given the sample $\boldsymbol{x}_t \in \mathbb{R}^d$, network's prediction $l$, and predicted label's pre-softmax score $S_l(\boldsymbol{x}_t)$, feature-importance methods assign an absolute score to each input dimension relative to its effect on score. Here we normalize the scores for each image by the sum of the scores to ensures that our attacks change not just the absolute feature saliencies, but their relative values.

**Simple gradient method** (Baehrens et al., 2010; Simonyan et al., 2013). Uses a linear approximation to detect the sensitivity of the score to perturbing each of the dimensions: $\boldsymbol{R}(\boldsymbol{x}_t)_j = |\nabla_{\boldsymbol{x}} S_l(\boldsymbol{x}_t)_j| / \sum_{i=1}^{d} |\nabla_{\boldsymbol{x}} S_l(\boldsymbol{x}_t)_i|$.

**Integrated gradients** To solve the saturation problem of simple gradient method, Sundararajan et al. (2017) introduced the integrated gradients method where given a reference input point $\boldsymbol{x}^0$, $\boldsymbol{R}(\boldsymbol{x}_t) = \left| \frac{\boldsymbol{x}_t - \boldsymbol{x}^0}{M} \sum_{k=1}^{M} \nabla_{\boldsymbol{x}} S_l \left( \frac{k}{M} (\boldsymbol{x}_t - \boldsymbol{x}^0) + \boldsymbol{x}^0 \right) \right|$, which is then normalized by its sum.

**DeepLIFT** Given a reference input $\boldsymbol{x}^0$, DeepLIFT decomposes the change in score $(S_l(\boldsymbol{x}_t) - S_l(\boldsymbol{x}_t)))$ backwards through the neural network. The score change from each layer is propagated to the previous layer proportional to its changes in the neuronal activations from the reference until the input layer is reached.We use DeepLIFT with the Rescale rule (Shrikumar et al., 2017)

### 2.2 Exemplar-Based Method: Influence Functions

Given *training examples*, $\{(\boldsymbol{x_i}, y_i)\}$ these methods find out which training examples, if up-weighted or down-weighted during training time, would have the biggest effect on the loss of the test input $(\boldsymbol{x_t}, y_t)$. Koh & Liang (2017) proposed the following as the influence function:

$$I(z_i, z_t) = -\nabla_\theta L(z_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z_i, \hat{\theta}), \tag{1}$$

where $z_i \stackrel{\text{def}}{=} (\boldsymbol{x}_i, y_i)$ and $L(z, \hat{\theta})$ is the loss of the network with parameters set to $\hat{\theta}$. $H_{\hat{\theta}} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \hat{\theta})$ is the empirical Hessian of the network calculated over the training examples.

### 2.3 Metrics for Interpretation Similarity

We use the **Spearman (1904) rank correlation** to compare interpretations and track how the relative importance of features/examples has changed. In addition, beccause in many settings only the dominant explanation is of interest, we also measure **top-k intersection**, defined to be the size of the intersection of $k$ most important features or examples.

## 3 Random and Systematic Perturbation Methods

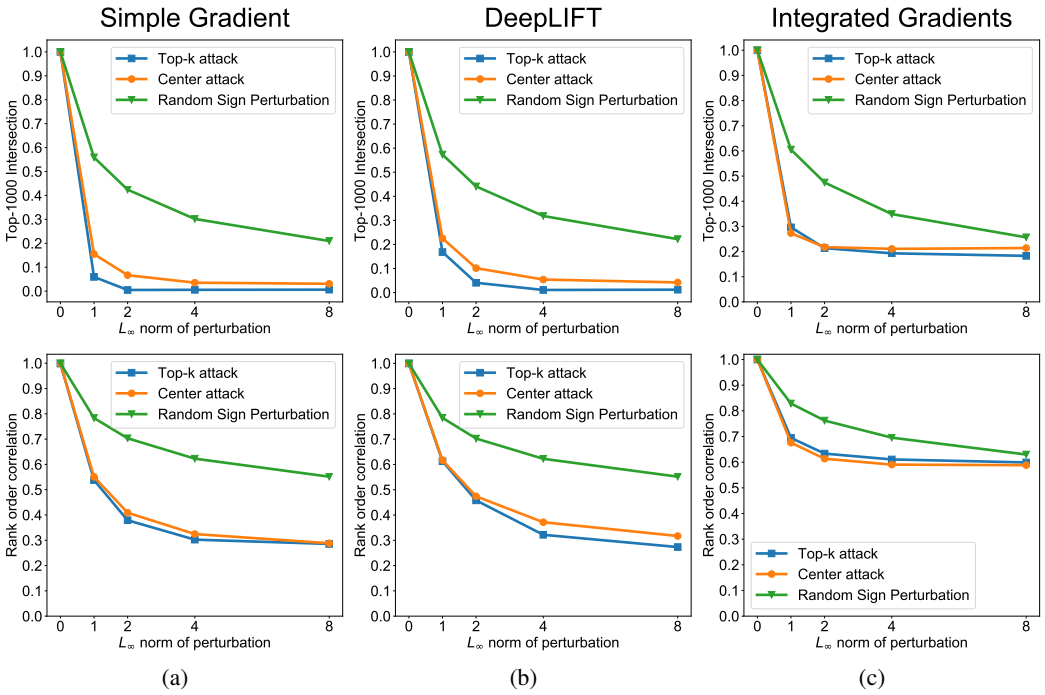
**Random sign perturbation** As baseline, we randomly perturb each pixel of the test image by $\pm\epsilon$.

**Iterative attacks against feature-importance methods** Described in Algorithm 1 in Appendix A, we define two adversarial attacks against feature-importance methods, each of which consists of iterative maximiziation of a differentiable interpretation dissimilarity function.

**Gradient sign attack against influence functions** We linearize (1) around the current input and parameters. Constraining the $\ell_\infty$ of perturbation to $\epsilon$, we obtain an optimal single-step perturbation:

$$\boldsymbol{\delta} = \epsilon \text{sign}(\nabla_{\boldsymbol{x}_t} I(z_i, z_t)) = -\epsilon \text{sign}(\nabla_{\boldsymbol{x}_t} \nabla_\theta L(z_t, \hat{\theta})^\top \underbrace{H_{\hat{\theta}}^{-1} \nabla_\theta L(z_i, \hat{\theta})}_{\text{independent of } \boldsymbol{x}_t}). \tag{2}$$

which is the direction to decrease the influence of the 3 most influential training images of the original test image[2]. Of course, this affects the influence of all of the other training images as well.[3]




(a) (b) (c)

Figure 2: **Comparison of adversarial attack algorithms on feature-importance methods.** Random sign perturbation already causes significant changes in both top-1000 intersection and rank order correlation. This suggests that the saliency of individual or small groups of pixels can be extremely fragile to the input and should be interpreted with caution. With targeted perturbations, we observe more dramatic fragility. Even with a perturbation of $L_\infty = 2$, the interpretations change significantly. CIFAR results are displayed in Appendix D.

## 4 Experiments & Results

The models and data sets we used are described in Appendix B.

**Results for feature-importance methods** Fig. 1 depicts examples of center-shift attack. More examples are displayed in Appendix C. Average results for 512 ImageNet test images are displayed in Fig. 2.

[2]In other words, we generate the perturbation given by: $-\epsilon \text{sign}(\sum_{i=1}^{3} \nabla_{\boldsymbol{x}_t} \nabla_\theta L(z_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z_{(i)}, \hat{\theta}))$, where $z_{(i)}$ is the $i^{\text{th}}$ most influential training image of the original test image.

[3]We followed same setup as the original work; the influence is only calculated with respect to the parameters that change during training. (Which are the parameters in the final layer of our network). See Section 4).

**Results for influence functions** Fig. 3 shows a representative test image to which we have applied the gradient sign attack. Although the prediction is the same, the most influential training examples change. Additional examples can be found in Appendix E. Average results for 200 images of roses and sunflowers are portrayed in Fig. 4.

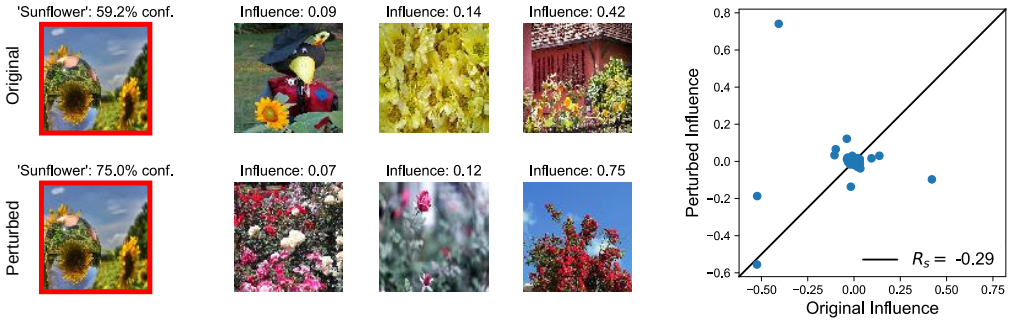
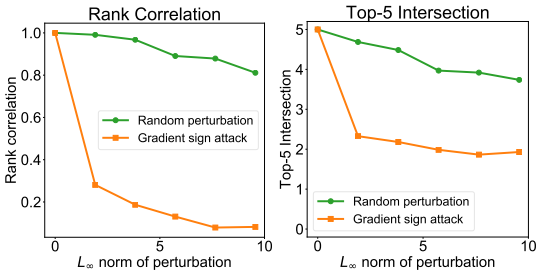



Figure 3: **Gradient sign attack on influence functions.** The top 3 training images identified by influence functions are shown in the top row. Using the gradient sign attack, we perturb the test image (with $\epsilon = 8$) to produce the leftmost image in the second row. The 3 most influential images (targeted by the attack) have decreased in influence, but the influences of other images have also changed.




(a) (b)

Figure 4: **Comparison of random and targeted perturbations on influence functions.** The effect of random attacks is small and generally doesn't affect the most influential images. A targeted attack, however, can significantly affect (a) the rank correlation and (b) even change the make-up of the 5 most influential images.

## 5 Conclusion

Our main message is that robustness of the interpretation of a prediction is an important and challenging problem, especially as in many applications (e.g. many biomedical and social settings) users are as interested in the interpretation as in the prediction itself. Our results raise concerns on how interpretation is sensitive to noise and can be manipulated. We do not suggest that interpretations are meaningless, just as adversarial attacks on predictions do not imply that neural networks are useless.

We show that these importance scores can be sensitive to even random perturbation. More dramatic manipulations of interpretations can be achieved with our targeted perturbations, which raise security concerns. A more thorough theoretical discussion could be found in Appendices F, H, G, and I. While we focus on image data (ImageNet and CIFAR-10), because these are the standard benchmarks for popular interpretation tools, this fragility issue can be wide-spread in biomedical, economic and other settings where neural networks are increasingly used. Understanding interpretation fragility in these applications and develop more robust methods are important agendas of research.

## References


David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pp. 854–863, 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *CoRR*, abs/1704.02685, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Charles Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72–101, 1904.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

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

# Appendices

## A Feature importance iterative attack algorithms

**Algorithm 1** Iterative Feature-Importance Attacks

**Input:** test image $\boldsymbol{x}_t$, maximum norm of perturbation $\epsilon$, normalized feature importance function $\boldsymbol{R}(\cdot)$, number of iterations $P$, step size $\alpha$

Define a dissimilarity function $D$ to measure the change between interpretations of two images:

$$D(\boldsymbol{x}_t, \boldsymbol{x}) = \begin{cases} -\sum\limits_{i \in B} \boldsymbol{R}(\boldsymbol{x})_i & \text{for } \textbf{top-k} \text{ attack} \\ \\ ||\boldsymbol{C}(\boldsymbol{x}) - \boldsymbol{C}(\boldsymbol{x}_t)||_2 & \text{for } \textbf{mass-center} \text{ attack,} \end{cases}$$

where $B$ is the set of the $k$ largest dimensions[a] of $\boldsymbol{R}(\boldsymbol{x}_t)$, and $\boldsymbol{C}(\cdot)$ is the center of saliency mass[b].

Initialize $\boldsymbol{x}^0 = \boldsymbol{x}_t$

**for** $p \in \{1, \ldots, P\}$ **do**

Perturb the test image in the direction of signed gradient[c] of the dissimilarity function:

$$\boldsymbol{x}^p = \boldsymbol{x}^{p-1} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} D(\boldsymbol{x}_t, \boldsymbol{x}^{p-1}))$$

If needed, clip the perturbed input to satisfy the norm constraint: $||\boldsymbol{x}^p - \boldsymbol{x}_t||_\infty \leq \epsilon$

**end for**

Among $\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^P\}$, return the element with the largest value for the dissimilarity function and the same prediction as the original test image.

[a] The goal is to damp the saliency scores of the $k$ features originally identified as the most important.

[b] The center of mass is defined for a $W \times H$ image as:

$$\boldsymbol{C}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{i \in \{1,\ldots,W\}} \sum_{j \in \{1,\ldots,H\}} \boldsymbol{R}(\boldsymbol{x})_{i,j}[i,j]^T$$

[c] In some networks, such as those with ReLUs, this gradient is always 0. To attack interpretability in such networks, we replace the ReLU activations with their smooth approximation (softplus) when calculating the gradient and generate the perturbed image using this approximation. The perturbed images that result are effective adversarial attacks against the original ReLU network, as discussed in Section 4.

## B Description of the CIFAR-10 classification network

For feature-importance methods ILSVRC2012 (ImageNet classification challenge data set) (Russakovsky et al., 2015) and CIFAR-10 (Krizhevsky, 2009) data sets were used. For the ImageNet classification data set, we used a pre-trained SqueezeNet[4] (Iandola et al., 2016) and for CIFAR-10 we trained our following network trained the using ADAM optimizer (Kingma & Ba, 2014) with default parameters. The resulting test accuracy using ReLU activation was 73%. For the experiment in Fig, 11(a), we replaced ReLU activation with Softplus and retrained the network (with the ReLU network weights as initial weights). The resulting accuracy was 73%.

[4] https://github.com/rcmalli/keras-squeezenet

| Network Layers |
|---|
| $3 \times 3$ conv. 96 ReLU |
| $3 \times 3$ conv. 96 ReLU |
| $3 \times 3$ conv. 96 Relu<br>Stride 2 |
| $3 \times 3$ conv. 192 ReLU |
| $3 \times 3$ conv. 192 ReLU |
| $3 \times 3$ conv. 192 Relu<br>Stride 2 |
| 1024 hidden sized feed forward |

We used pixel-wise and the channel-wise mean images as the CIFAR-10 and ImageNet reference points respectively.[5] We ran all iterative attack algorithms for $P = 300$ iterations with step size $\alpha = 0.5$.

To evaluate the robustness of influence functions, we followed a similar experimental setup to that of the original authors: we trained an InceptionNet v3 with all but the last layer frozen (the weights were pre-trained on ImageNet and obtained from Keras[6]). The last layer was trained on a binary flower classification task (**roses** vs. **sunflowers**), using a data set consisting of 1,000 training images[7]. This data set was chosen because it consisted of images that the network had not seen during pre-training on ImageNet. The network achieved a validation accuracy of 97.5% on this task.

## C Additional examples of feature importance perturbations

Here we provide three more examples from ImageNet. For each example, all three methods of feature importance are attacked by random sign noise and our two targeted adversarial algorithms.

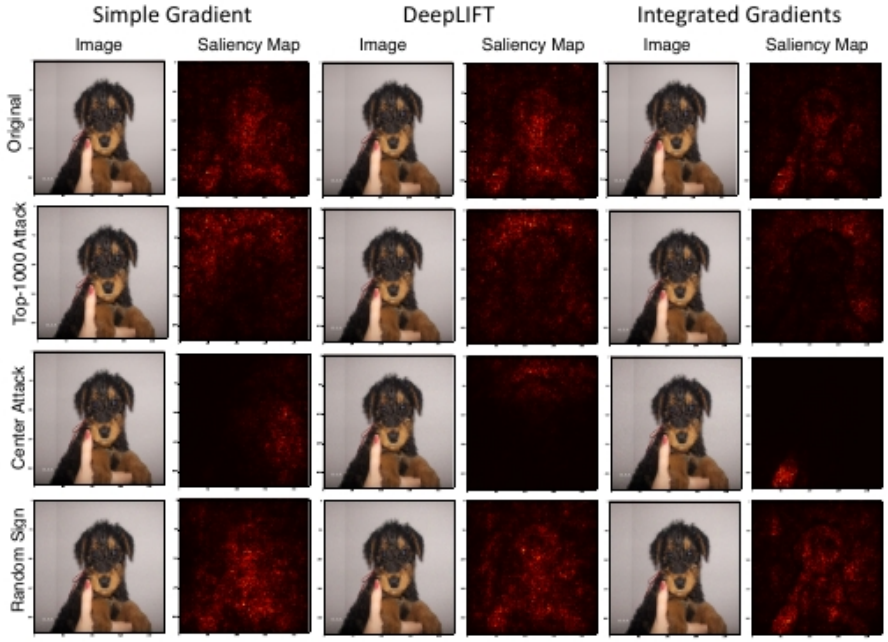



Figure 5: All of the images are classified as a *airedale*.

[5] For the integrated gradients method we used parameter M=100.

[6] https://keras.io/applications/

[7] adapted from: https://www.tensorflow.org/tutorials/image_retraining

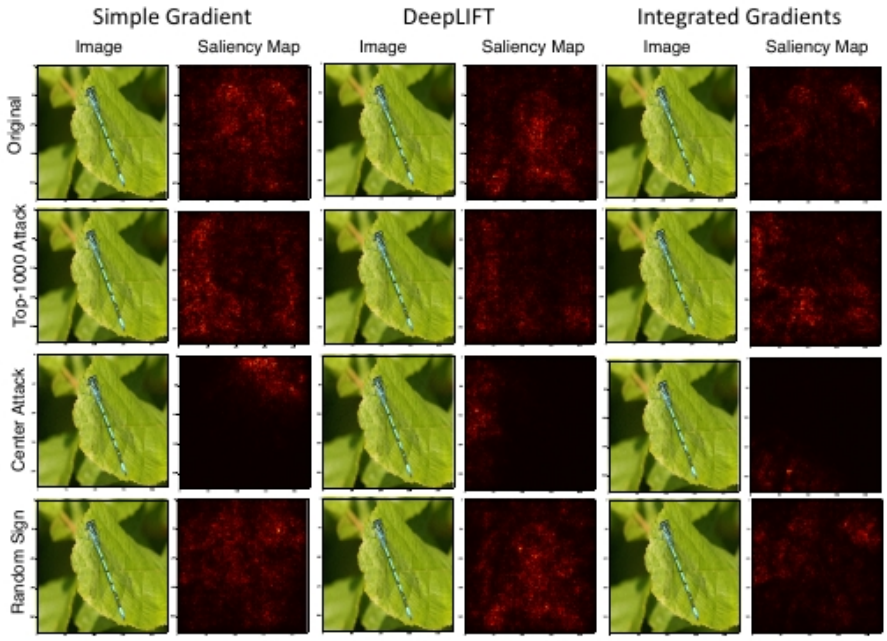



Figure 6: All of the images are classified as a *damselfly*.

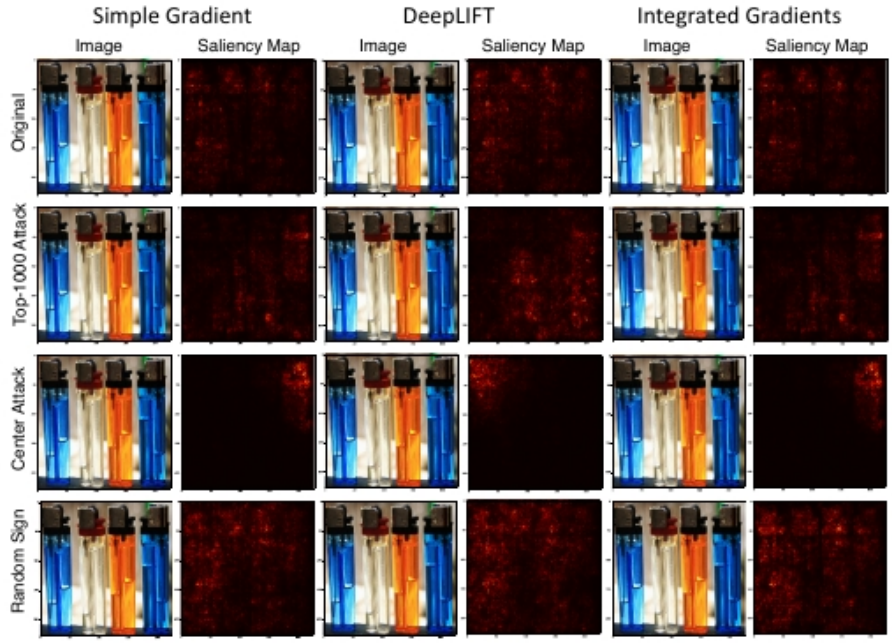



Figure 7: All of the images are classified as a *lighter*.

## D Measuring center of mass movement

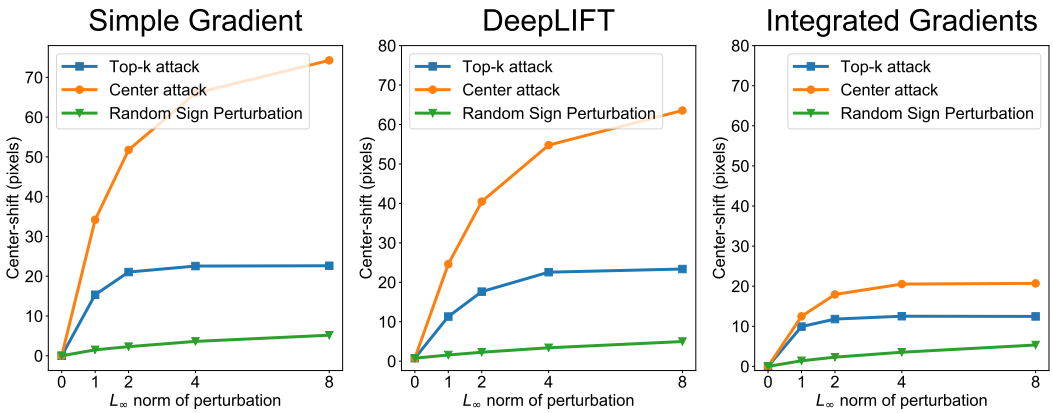



Figure 8: **Center-shift results for three feature importance methods on ImageNet**: As discussed in the paper, among our three measurements, center-shift measure was the most correlated measure with the subjective perception of change in saliency maps. The results in Appendix C also show that the center attack which resulted in largest average center-shift, also results in the most significant subjective change in saliency maps. Random sign perturbations, on the other side, did not substantially change the global shape of the saliency maps, though local pockets of saliency are sensitive. Just like rank correlation and top-1000 intersection measures, the integrated gradients method is the most robust method against adversarial attacks in the center-shift measure .

**Results for adversarial attacks against CIFAR-10 feature importance methods**

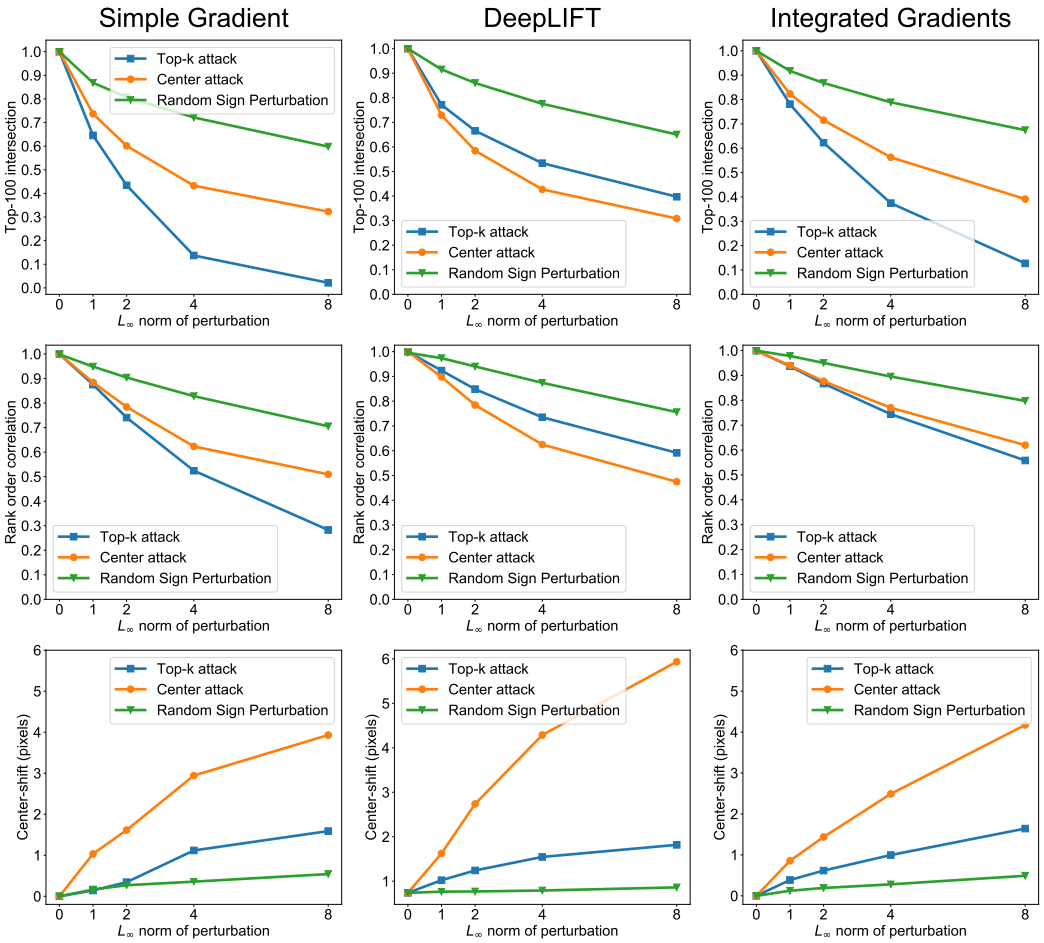
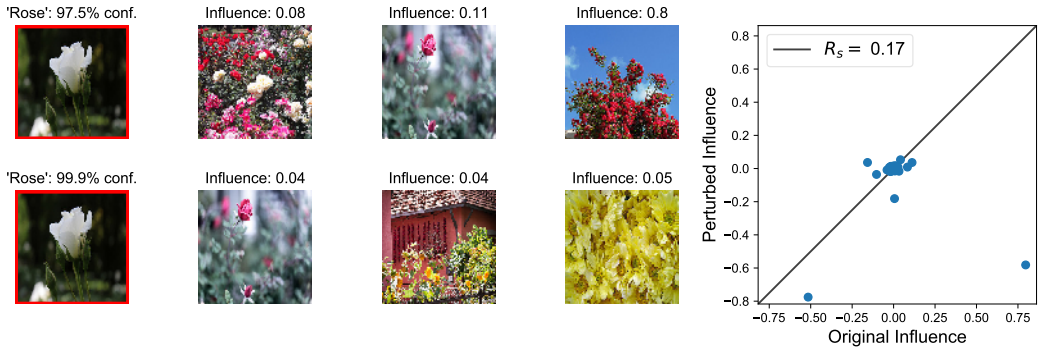


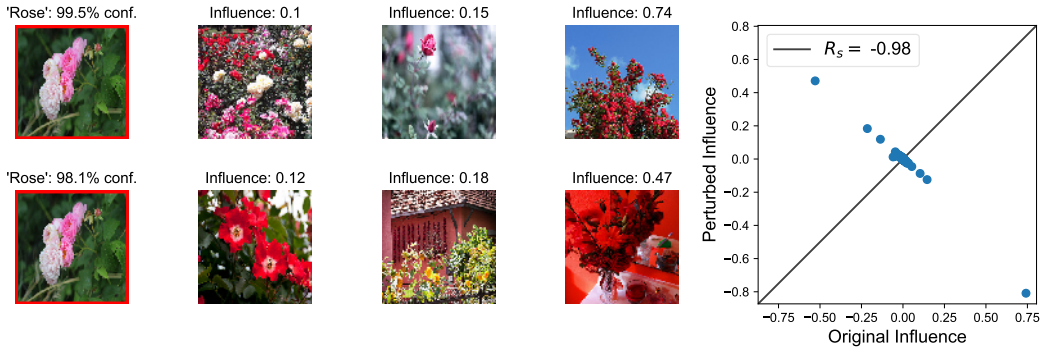

Figure 9: **Results for adversarial attacks against CIFAR10 feature importance methods**: Using750 CIFAR10 test images, the center-shift attack and top-k attack with k=100 achieve similar results for rank correlation and top-100 intersection measurements and both are stronger than random perturbations. Center-shift attack moves the center of mass more than two other perturbations. Among different feature importance methods, integrated gradients is more robust than the two other methods. Additionally, results for CIFAR10 show that images in this data set are more robust against adversarial attack compared to ImageNet images which agrees with our analysis that higher dimensional inputs are tend to be more fragile.

## E Additional Examples of Adversarial Attacks on Influence Functions

In this appendix, we provide additional examples of the fragility of influence functions, analogous to Fig. 3.




(a)




(b)

Figure 10: **Further examples of gradient-sign attacks on influence functions.** (a) Here we see a representative example of the most influential training images before and after a perturbation to the test image. The most influential image before the attack is one of the least influential afterwards. Overall, the influences of the training images before and after the attack are uncorrelated. (b) In this example, the perturbation has remarkably caused the training images to almost completely reverse in influence. Training images that had the most positive effect on prediction now have the most negative effects and the other way round.

## F Hessian analysis

In this section, we try to understand the source of interpretation fragility. The question is whether fragility a consequence of the complex non-linearities of a deep network or a characteristic present even in high-dimensional linear models, as is the case for adversarial examples for prediction (Goodfellow et al., 2014). To gain more insight into the fragility of gradient based interpretations, let $S(\boldsymbol{x}; \boldsymbol{W})$ denote the score function of interest; $\boldsymbol{x} \in \mathbb{R}^d$ is an input vector and $\boldsymbol{W}$ is the weights of the neural network, which is fixed since the network has finished training. We are interested in the Hessian $H$ whose entries are $H_{i,j} = \frac{\partial S}{\partial x_i \partial x_j}$. The reason is that the first order approximation of gradient for some input perturbation direction $\boldsymbol{\delta} \in \mathbb{R}^d$ is: $\nabla_{\boldsymbol{x}} S(\boldsymbol{x} + \boldsymbol{\delta}) - \nabla_{\boldsymbol{x}} S(\boldsymbol{x}) \approx H\boldsymbol{\delta}$.

First, consider a linear model whose score for an input $\boldsymbol{x}$ is $S = \boldsymbol{w}^\top \boldsymbol{x}$. Here, $\nabla_{\boldsymbol{x}} S = \boldsymbol{w}$ and $\nabla^2_{\boldsymbol{x}} S = 0$; the feature-importance vector $\boldsymbol{w}$ is robust, because it is completely independent of $\boldsymbol{x}$. Thus, some non-linearity is required for interpretation fragility. A simple network that is susceptible to adversarial attacks on interpretations consists of a set of weights connecting the input to a single neuron followed by a non-linearity (e.g. logistic regression): $S = g(\boldsymbol{w}^\top \boldsymbol{x})$.

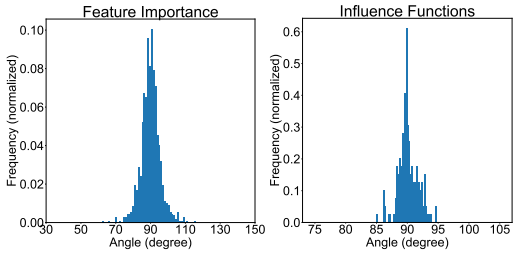
We can calculate the change in saliency map due to a small perturbation in $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\delta}$. The first-order approximation for the change in saliency map will be equal to : $H \cdot \boldsymbol{\delta} = \nabla^2_{\boldsymbol{x}} S \cdot \boldsymbol{\delta}$. In particular, the saliency of the $i^{\text{th}}$ feature changes by $(\nabla^2_{\boldsymbol{x}} S \cdot \boldsymbol{\delta})_i$ and furthermore, the relative change

is $(\nabla^2_{\boldsymbol{x}} S \cdot \boldsymbol{\delta})_i / (\nabla_{\boldsymbol{x}} S)_i$. For the simple network, this relative change is:

$$\frac{(\boldsymbol{w}\boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x}))_i}{(\boldsymbol{w} g'(\boldsymbol{w}^\top \boldsymbol{x}))_i} = \frac{w_i \boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x})}{w_i g'(\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x})}{g'(\boldsymbol{w}^\top \boldsymbol{x})}, \tag{3}$$

where we have used $g'(\cdot)$ and $g''(\cdot)$ to refer to the first and second derivatives of $g(\cdot)$. Note that $g'(\boldsymbol{w}^\top \boldsymbol{x})$ and $g''(\boldsymbol{w}^\top \boldsymbol{x})$ do not scale with the dimensionality of $\boldsymbol{x}$ because in general, independent from the dimensionality, $\boldsymbol{x}$ and $\boldsymbol{w}$ are $\ell_2$-normalized or have fixed $\ell_2$-norm due to data preprocessing and weight decay regularization. However, if we choose $\boldsymbol{\delta} = \epsilon \text{sign}(\boldsymbol{w})$, then the relative change in the saliency grows with the dimension, since it is proportional to the $\ell_1$-norm of $\boldsymbol{w}$. When the input is high-dimensional—which is the case with images—the relative effect of the perturbation can be substantial. Note also that this perturbation is exactly the sign of the first right singular vector of the Hessian $\nabla^2_{\boldsymbol{x}} S$, which is appropriate since that is the vector that has the maximum effect on the gradient of $S$. A similar analysis can be carried out for influence functions (see Appendix G).

For this simple network, the direction of adversarial attack on interpretability, $\text{sign}(\boldsymbol{w})$ is the same as the adversarial attack on prediction. This means that we cannot perturb interpretability independently of prediction. For more complex networks, this is not the case and in Appendix H we show this analytically for a simple case of a two-layer network. As an empirical test, in Fig. 11(a), we plot the distribution of the angle between $\nabla_{\boldsymbol{x}} S$ and $\boldsymbol{v}_1$ (the first right singular vector of $H$ which is the most fragile direction of feature importance) for 1000 CIFAR10 images (Details of the network in Appendix B). In Fig. 11(b), we plot the equivalent distribution for influence functions, computed across all 200 test images. The result confirms that the steepest direction of change in interpretation and prediction are generally orthogonal, justifying how the perturbations can change the interpretation without changing the prediction.




(a) (b)

Figure 11: **Orthogonality of Prediction and Interpretation Fragile Directions** (a) The histogram of the angle between the steepest direction of change in feature importance and the steepest score change direction. (b) The distribution of the angle between the gradient of the loss function and the steepest direction of change of influence of the most influential image.

## G Dimensionality-Based Explanation for Fragility of Influence Functions

Here, we demonstrate that increasing the dimension of the input of a simple neural network increases the fragility of that network with respect to influence functions, analogous to the calculations carried out for importance-feature methods in Section **??**. Recall that the influence of a training image $z_i = (\boldsymbol{x}_i, y_i)$ on a test image $z = (\boldsymbol{x}, y)$ is given by:

$$I(z_i, z) = -\underbrace{\nabla_\theta L(z, \hat{\theta})^\top}_{\text{dependent on } \boldsymbol{x}} \underbrace{H_{\hat{\theta}}^{-1} \nabla_\theta L(z_i, \hat{\theta})}_{\text{independent of } \boldsymbol{x}}. \tag{4}$$

We restrict our attention to the term in (4) that is dependent on $\boldsymbol{x}$, and denote it by $J \stackrel{\text{def}}{=} \nabla_\theta L$. $J$ represents the infinitesimal effect of each of the parameters in the network on the loss function evaluated at the test image.

Now, let us calculate the change in this term due to a small perturbation in $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\delta}$. The first-order approximation for the change in $J$ is equal to: $\nabla_{\boldsymbol{x}} J \cdot \boldsymbol{\delta} = \nabla_\theta \nabla_{\boldsymbol{x}} L \cdot \boldsymbol{\delta}$. In particular, for the $i^{\text{th}}$ parameter, $J_i$ changes by $(\nabla_\theta \nabla_{\boldsymbol{x}} L \cdot \boldsymbol{\delta})_i$ and furthermore, the relative change is $(\nabla_\theta \nabla_{\boldsymbol{x}} L \cdot \boldsymbol{\delta})_i / (\nabla_\theta L)_i$. For the simple network defined in Section **??**, this evaluates to (replacing $\theta$ with $\boldsymbol{w}$ for consistency of notation):

$$\frac{(\boldsymbol{x}\boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x}))_i}{(\boldsymbol{x} g'(\boldsymbol{w}^\top \boldsymbol{x}))_i} = \frac{x_i \boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x})}{x_i g'(\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x})}{g'(\boldsymbol{w}^\top \boldsymbol{x})}, \tag{5}$$

where for simplicity, we have taken the loss to be $L = |y - g(\boldsymbol{w}^\top x)|$, making the derivatives easier to calculate. Furthermore, we have used $g'(\cdot)$ and $g''(\cdot)$ to refer to the first and second derivatives of $g(\cdot)$. Note that $g'(\boldsymbol{w}^\top \boldsymbol{x})$ and $g''(\boldsymbol{w}^\top \boldsymbol{x})$ do not scale with the dimensionality of $\boldsymbol{x}$ because $\boldsymbol{x}$ and $\boldsymbol{w}$ are generalized $L_2$-normalized due to data preprocessing and weight decay regularization.

However, if we choose $\boldsymbol{\delta} = \epsilon \text{sign}(\boldsymbol{w})$, then the relative change in the saliency grows with the dimension, since it is proportional to the $L_1$-norm of $\boldsymbol{w}$.

# H Orthogonality of steepest directions of change in score and feature importance functions in a Simple Two-layer network

Consider a two layer neural network with activation function $g(\cdot)$, input $\boldsymbol{x} \in \mathbb{R}^d$, hidden vector $\boldsymbol{u} \in \mathbb{R}^h$ , and score function $S$), we have:

$$S = \boldsymbol{v} \cdot \boldsymbol{u} = \sum_{j=1}^{h} v_j u_j$$

$$\boldsymbol{u} = g(W^T \boldsymbol{x}) \rightarrow u_j = \boldsymbol{w}_j.\boldsymbol{x}$$

where $\boldsymbol{w}_j = ||\boldsymbol{w}_j||_2 \hat{\boldsymbol{w}}_j$. We have:

$$\nabla_{\boldsymbol{x}} S = \sum_{j=1}^{h} v_j \nabla_{\boldsymbol{x}} u_j = \sum_{j=1}^{h} v_j g^{'}(\boldsymbol{w}_j.\boldsymbol{x}) \boldsymbol{w}_j$$

$$\nabla^2_{\boldsymbol{x}} S = \sum_{j=1}^{h} v_j \nabla^2_{\boldsymbol{x}} u_j = \sum_{j=1}^{h} v_j g^{''}(\boldsymbol{w}_j.\boldsymbol{x}) \boldsymbol{w}_j^T \boldsymbol{w}_j$$

Now for an input sample $\boldsymbol{x}$ perturbation $\boldsymbol{\delta}$, for the change in feature importance:

$$\nabla_{\boldsymbol{x}} S(\boldsymbol{x} + \boldsymbol{\delta}) - \nabla_{\boldsymbol{x}} S(\boldsymbol{x}) \approx \nabla^2_{\boldsymbol{x}} S \cdot \boldsymbol{\delta}$$

which is equal to:

$$\sum_{j=1}^{h} v_j g^{''}(\boldsymbol{w}_j.\boldsymbol{x})(\boldsymbol{w}_j \cdot \boldsymbol{\delta}) \boldsymbol{w}_j$$

We further assume that the input is high-dimensional so that $h < d$ and for $i \neq j$ we have $\boldsymbol{w}_j \cdot \boldsymbol{w}_i = 0$. For maximizing the $\ell_2$ norm of saliency difference we have the following perturbation direction:

$$\boldsymbol{\delta}_m = \text{argmax}_{||\boldsymbol{\delta}||=1} ||\nabla_{\boldsymbol{x}} S(\boldsymbol{x} + \boldsymbol{\delta}) - \nabla_{\boldsymbol{x}} S(\boldsymbol{x})|| = \hat{\boldsymbol{w}}_k$$

where:

$$k = \text{argmax} |v_j g^{''}(\boldsymbol{w}_j.\boldsymbol{x})| \times ||\boldsymbol{w}_k||_2^2$$

comparing which to the direction of feature importance:

$$\frac{\nabla_{\boldsymbol{x}} S(\boldsymbol{x})}{||\nabla_{\boldsymbol{x}} S(\boldsymbol{x})||_2} = \sum_{i=1}^{h} \frac{v_j g^{'}(\boldsymbol{w}_i \cdot \boldsymbol{x}) ||\boldsymbol{w}_i||_2}{(\sum_{j=1}^{h} v_j g^{'}(\boldsymbol{w}_j \cdot \boldsymbol{x}) ||\boldsymbol{w}_j||_2)^2} \hat{\boldsymbol{w}}_i$$

we conclude that the two directions are not parallel unless $g^{'}(.) = g^{''}(.)$ which is not the case for many activation functions like Softplus, Sigmoid, etc.

# I Designing Interpretability-Robust Networks

The analyses and experiments in this paper have demonstrated that small perturbations in the input layers of deep neural networks can have large changes in the interpretations. This is analogous to classical adversarial examples, whereby small perturbations in the input produce large changes in the *prediction*. In that setting, it has been proposed that the Lipschitz constant of the network be constrained during training to limit the effect of adversarial perturbations (Szegedy et al., 2013). This has found some empirical success (Cisse et al., 2017).

Here, we propose an analogous method to upper-bound the change in interpretability of a neural network as a result of perturbations to the input. Specifically, consider a network with $K$ layers, which takes as input a data point we denote as $y_0$. The output of the $i^{\text{th}}$ layer is given by $y_{i+1} = f_i(y_i)$ for $i = 0, 1 \dots K-1$. We define $S \stackrel{\text{def}}{=} f_{K-1}(f_{K-2}(\dots f_0(y_0)\dots))$ to be the output (e.g. score for the correct class) of our network, and we are interested in designing a network whose gradient $S' = \nabla_{y_0} S$ is relatively insensitive to perturbations in the input, as this corresponds to a network whose feature importances are robust.

A natural quantity to consider is the Lipschitz constant of $S'$ with respect to $y_0$. By the chain rule, the Lipschitz constant of $S'$ is

$$\mathcal{L}(S') = \mathcal{L}(\frac{\delta y_k}{\delta y_{k-1}}) \dots \mathcal{L}(\frac{\delta y_1}{\delta y_0}) \tag{6}$$

Now consider the function $f_i(\cdot)$, which maps $y_i$ to $y_{i+1}$. In the simple case of the fully-connected network, which we consider here, $f_i(y_i) = g_i(W_i y_i)$, where $g_i$ is a non-linearity and $W_i$ are the trained weights for that layer. Thus, the Lipschitz constant of the $i^{\text{th}}$ partial derivative in (6) is the Lipschitz constant of

$$\frac{\delta f_i}{\delta y_{i-1}} = W_i g_i'(W_i y_{i-1}),$$

which is upper-bounded by $||W_i||^2 \cdot \mathcal{L}(g_i'(\cdot))$, where $||W||$ denotes the operator norm of $W$ (its largest singular value)[8]. This suggests that a conservative upper ceiling for (6) is

$$\mathcal{L}(S') \leq \prod_{i=0}^{K-1} ||W_i||^2 \mathcal{L}(g_i'(\cdot)) \tag{7}$$

Because the Lipschitz constant of the non-linearities $g_i'(\cdot)$ are fixed, this result suggests that a regularization based on the operator norms of the weights $W_i$ may allow us to train networks that are robust to attacks on feature importance. The calculations in this Appendix section is meant to be suggestive rather than conclusive, since in practice the Lipschitz bounds are rarely tight.

[8] this bound follows from the fact that the Lipschitz constant of the composition of two functions is the product of their Lipschitz constants, and the Lipschitz constant of the product of two functions is also the product of their Lipschitz constants.