INTERPRETATION OF NEURAL NETWORK IS FRAGILE

Anonymous authors
Paper under double-blind review

ABSTRACT

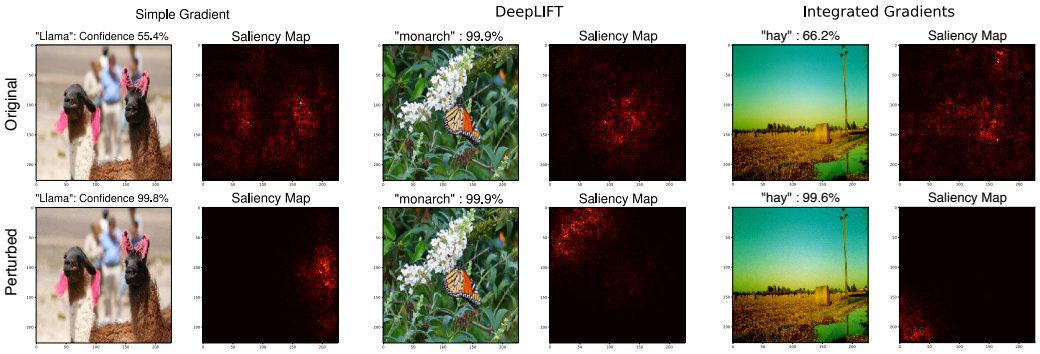
In order for machine learning to be deployed and trusted in many applications, it is crucial to be able to reliably explain why the machine learning algorithm makes certain predictions. For example, if an algorithm classifies a given pathology image to be a malignant tumor, then the doctor may need to know which parts of the image led the algorithm to this classification. How to interpret black-box predictors is thus an important and active area of research. 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 following sense: 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 interpretation methods (saliency maps, integrated gradient, and DeepLIFT) on ImageNet and CIFAR-10. Our experiments show that even small random perturbation can change the feature importance and new systematic perturbations can lead to dramatically different interpretations without changing the label. We extend these results to show that interpretations based on exemplars (e.g. influence functions) are similarly fragile. Our analysis of the geometry of the Hessian matrix gives insight on why fragility could be a fundamental challenge to the current interpretation approaches.

1 INTRODUCTION

Predictions made by machine learning algorithms play an important role in our everyday lives and can affect decisions in technology, medicine, and even the legal system (Rich, 2015; Obermeyer & Emanuel, 2016). As the 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. Similarly, if an algorithm predicts an individual to be a credit risk, then the lender (and the borrower) might want to know why. Therefore having interpretations for why certain predictions are made is critical for establishing trust and transparency between the users and the algorithm (Lipton, 2016).

Having an interpretation is not enough, however. The explanation itself must be robust in order to establish human trust. Take the pathology predictor; an interpretation method might suggest that a particular section in an image is important for the malignant classification (e.g. that section could have high scores in saliency map). The clinician might then focus on that section for investigation, treatment or even look for similar features in other patients. It would be highly disconcerting if in an extremely similar image, visually indistinguishable from the original and also classified as malignant, a very different section is interpreted as being salient for the prediction. 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



(a) (b) (c)

Figure 1: **The fragility of feature-importance maps.** We generate feature-importance scores, also called saliency maps, using three popular interpretation methods: simple gradient (a), DeepLIFT (b) and integrated gradient (c). The **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; in fact the network's (SqueezeNet) confidence in the prediction has actually increased. However, the saliency maps of the perturbed images are meaningless.

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).

Our findings highlight the fragility of popular interpretation methods, which has not been carefully considered in literature. Fragility directly limits how much we can trust and learn from the interpretations. It also raises a significant new security concern. Especially in medical or economic applications, users often take the interpretation of a prediction as containing causal insight (*"this image is a malignant tumor likely because of the section with a high saliency score"*). An adversary could minutely manipute the input to draw attention away from relevant features or onto his/her desired features. Such attacks might be especially hard to detect as the actual labels have not changed.

While we focus on image data here because most of the interpretation methods have been motivated by images, the fragility of interpretation could be a much broader problem. Fig. 2 illustrates the intuition that when the decision boundary in the input feature space is complex, as is the case with deep nets, a small perturbation in the input can push the example into a region with very different loss contours. Because the feature importance is closely related to the gradient which is perpendicular to the loss contours, the importance scores can also be dramatically different. We provide additional analysis of this in Section 5.

2 Interpretation Methods for Neural Network Predictions

2.1 Feature-Importance Interpretation

This first class of methods explains predictions in terms of the relative importance of features in a test input sample. Given the sample $\boldsymbol{x}_t \in \mathbb{R}^d$ and the network's prediction l, we define the score of the predicted class $S_l(\boldsymbol{x}_t)$ to be the value of the l-th output neuron right before the softmax operation. We take l to be the class with the max score; i.e. the predicted class. Feature-importance methods seek to find the dimensions of input data point that most strongly affect the score, and in doing so, these methods assign an absolute saliency score to each input feature. Here we normalize the scores for each image by the sum of the saliency scores across the features. This ensures that any perturbations that we design change not the absolute feature saliencies (which may still preserve the ranking of different features), but their relative values. We summarize three different methods to calculate the normalized saliency score, denoted by $\boldsymbol{R}(\boldsymbol{x}_t)$.

Simple gradient method Introduced in Baehrens et al. (2010) and applied to deep neural networks in Simonyan et al. (2013), the simple gradient method applies a local linear approximation of the model to detect the sensitivity of the score to perturbing each of the input dimensions. Given

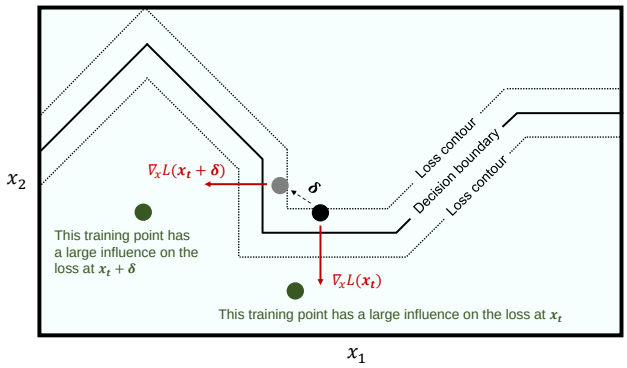


Figure 2: **Intuition for why interpretation is fragile.** Consider a test example $\boldsymbol{x_t} \in \mathbb{R}^2$ (black dot) that is slightly perturbed to a new position $\boldsymbol{x_t} + \boldsymbol{\delta}$ in input space (gray dot). The contours and decision boundary corresponding to a loss function (L) for a two-class classification task are also shown, allowing one to see the direction of the gradient of the loss with respect to the input space. Neural networks with many parameters have decision boundaries that are roughly piecewise linear with many transitions. We illustrate that points near the transitions are especially fragile to interpretability-based analysis. A small perturbation to the input changes the direction of $\nabla_x L$ from being in the direction of x_1 to being in the direction of x_2, directly affecting feature-importance analyses. Similarly, a small perturbation to the test image changes which training image, when up-weighted, has the largest influence on L, directly affecting exemplar-based analysis.

input $\boldsymbol{x}_t \in \mathbb{R}^d$, the score is defined as:

$$\boldsymbol{R}(\boldsymbol{x}_t)_j = |\nabla_{\boldsymbol{x}} S_l(\boldsymbol{x}_t)_j| / \sum_{i=1}^{d} |\nabla_{\boldsymbol{x}} S_l(\boldsymbol{x}_t)_i|. \tag{1}$$

Integrated gradients A significant drawback of the simple gradient method is the saturation problem discussed by Shrikumar et al. (2017); Sundararajan et al. (2017). Consequently, Sundararajan et al. (2017) introduced the integrated gradients method where the gradients of the score with respect to M scaled versions of the input are summed and then multiplied by the input. Letting $\boldsymbol{x}^0$ be the reference point and $\Delta \boldsymbol{x}_t = \boldsymbol{x}_t - \boldsymbol{x}^0$, the feature importance vector is calculated by:

$$\boldsymbol{R}(\boldsymbol{x}_t) = \left| \frac{\Delta \boldsymbol{x}_t}{M} \sum_{k=1}^{M} \nabla_{\boldsymbol{x}} S_l \left(\frac{k}{M} \Delta \boldsymbol{x}_t + \boldsymbol{x}^0 \right) \right|, \tag{2}$$

which is then normalized for our analysis. Here the absolute value is taken for each dimension.

DeepLIFT DeepLIFT is an improved version of layer-wise relevance propagation (LRP) method (Bach et al., 2015). LRP methods decompose the score $S_l(\boldsymbol{x}_t)$ backwards through the neural network. In each step, the score from the last layer is propagated to the previous layer, with the score being divided proportionally to magnitude of the activations of the neurons in the previous layer. The scores are propagated to the input layer, and the result is a relevance score assigned to each of the input dimensions. DeepLIFT (Shrikumar et al., 2017) defines a reference point in the input space and propagates relevance scores proportionally to the changes in the neuronal activations from the reference. We use DeepLIFT with the Rescale rule; see Shrikumar et al. (2017) for details.

2.2 Exemplar-Based Methods: Influence Functions

A complementary approach to interpreting the results of a neural network is to explain the prediction of the network in terms of its *training examples*, $\{(\boldsymbol{x_i}, y_i)\}$. Specifically, we can ask: which training examples, if up-weighted or down-weighted during training time, would have the biggest effect on the loss of the test example $(\boldsymbol{x_t}, y_t)$? Koh & Liang (2017) proposed a method to calculate this value, called the influence, defined by the following equation:

$$I(z_i, z_t) = -\nabla_\theta L(z_t, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z_i, \hat{\theta}), \tag{3}$$

where $z_i \stackrel{\text{def}}{=} (\boldsymbol{x}_i, y_i)$ and z_t is defined analogously. $L(z, \hat{\theta})$ is the loss of the network with parameters set to $\hat{\theta}$ for the (training or test) data point z. $H_{\hat{\theta}} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \hat{\theta})$ is the empirical Hessian of the network calculated over the training examples. The training examples with the highest influence are understood as explaining *why* a network made a particular prediction for a test example.

2.3 METRICS FOR INTERPRETATION SIMILARITY

We consider two natural metrics for quantifying the similarity between interpretations for two different images.

- **Spearman's rank order correlation**: Because interpretation methods rank all of the features or training examples in order of importance, it natural to use the rank correlation (Spearman, 1904) to compare the similarity between interpretations.
- **Top-k intersection**: In many settings, only the most important features or interpretations are of interest. In these settings, we can compute the size of the intersection of the k most important features before and after perturbation.

3 RANDOM AND SYSTEMATIC PERTURBATIONS

Problem statement For a given fixed neural network $\mathscr{N}$ and input data point $\boldsymbol{x}_t$, the feature importance and influence function methods that we have described produce an interpretation $\boldsymbol{I}(\boldsymbol{x}_t; \mathscr{N})$. For feature importance, $\boldsymbol{I}(\boldsymbol{x}_t; \mathscr{N})$ is a vector of feature scores; for influence function $\boldsymbol{I}(\boldsymbol{x}_t; \mathscr{N})$ is a vector of scores for training examples. We would like to devise efficient perturbations to change the interpretability of a test image. Yet, the perturbations should be visually imperceptible and should not change the label of the prediction. Formally, we define the problem as:

$$\arg\max_{\boldsymbol{\delta}} \mathcal{D}\left(\boldsymbol{I}(\boldsymbol{x}_t; \mathscr{N}), \boldsymbol{I}(\boldsymbol{x}_t + \boldsymbol{\delta}; \mathscr{N})\right)$$
$$\text{subject to: } ||\boldsymbol{\delta}||_\infty \leq \epsilon, \text{Prediction}(\boldsymbol{x}_t + \boldsymbol{\delta}; \mathscr{N}) = \text{Prediction}(\boldsymbol{x}_t; \mathscr{N})$$

where $\mathcal{D}(\cdot)$ measures the change in interpretation (e.g. how many of the top-k pixels are no longer the top-k pixels of the saliency map after the perturbation) and $\epsilon > 0$ constrains the norm of the perturbation. In this paper, we carry out three kinds of input perturbations.

Random sign perturbation As a baseline, we generate random perturbations in which each pixel is randomly perturbed by $\pm\epsilon$. This is used to measure robustness against untargeted perturbations.

Iterative attacks against feature-importance methods In Algorithm 1 we define two adversarial attacks against feature-importance methods, each of which consists of taking a series of steps in the direction that maximizes a differentiable dissimilarity function between the original and perturbed interpretation. (1) The **top-k** attack seeks to perturb the saliency map by decreasing the relative importance of the k most important features of the original image. (2) When the input data are images, the center of mass of the saliency map often captures the user's attention. The **mass-center** attack is designed to result in the maximum spatial displacement of the center of mass of the saliency scores. Both of these attacks can be applied to any of the three feature-importance methods.

Gradient sign attack against influence functions We can obtain effective adversarial images for influence functions without resorting to interative procedures. We linearize (3) around the values of the current inputs and parameters. If we further constrain the L_∞ norm of the perturbation to ϵ, we obtain an optimal single-step perturbation:

$$\boldsymbol{\delta} = \epsilon \text{sign}(\nabla_{\boldsymbol{x}_t} I(z_i, z_t)) = -\epsilon \text{sign}(\nabla_{\boldsymbol{x}_t} \nabla_\theta L(z_t, \hat{\theta})^\top \underbrace{H_{\hat{\theta}}^{-1} \nabla_\theta L(z_i, \hat{\theta})}_{\text{independent of } \boldsymbol{x}_t}). \tag{4}$$

This perturbation can be applied to the pixels of a test image $x_t \to x_t + \delta$ to increase or decrease the influence of a particular training example, x_i. The attack we use consists of applying the negative of the perturbation in (4) to decrease the influence of the 3 most influential training images of the original test image[1]. Of course, this affects the influence of all of the other training images as well.

[1] 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 ith most influential training image of the original test image.

Algorithm 1 Iterative Feature-Importance Attacks

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

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

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

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.

We follow the same setup for computing the influence function as was done by the authors of Koh & Liang (2017). Because the influence is only calculated with respect to the parameters that change during training, we calculate the gradients only with respect to parameters in the final layer of our network (InceptionNet, see Section 4). This makes it feasible for us to compute (4) exactly, but it gives us the perturbation of the input *into the final layer*, not the first layer. So, we use standard back-propagation to calculate the corresponding gradient for the input test image. We then take the sign of this gradient as the perturbation and clip the image to produce the adversarial test image.

4 Experiments & Results

Data sets and models To evaluate the robustness of feature-importance methods, we used two image classification data sets: ILSVRC2012 (ImageNet classification challenge data set) (Russakovsky et al., 2015) and CIFAR-10 (Krizhevsky, 2009). For the ImageNet classification data set, we used a pre-trained SqueezeNet[2] model introduced by Iandola et al. (2016). For the CIFAR-10 data set we trained our own convolutional network, whose architecture is presented in Appendix A.

For both data sets, the results are examined using simple gradient, integrated gradients, and DeepLIFT feature importance methods. For DeepLIFT, we used the pixel-wise and the channel-wise mean images as the CIFAR-10 and ImageNet reference points respectively. For the integrated gradients method, the same references were used with parameter M=100. We ran all iterative attack algorithms for $P = 300$ iterations with step size $\alpha = 0.5$.

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

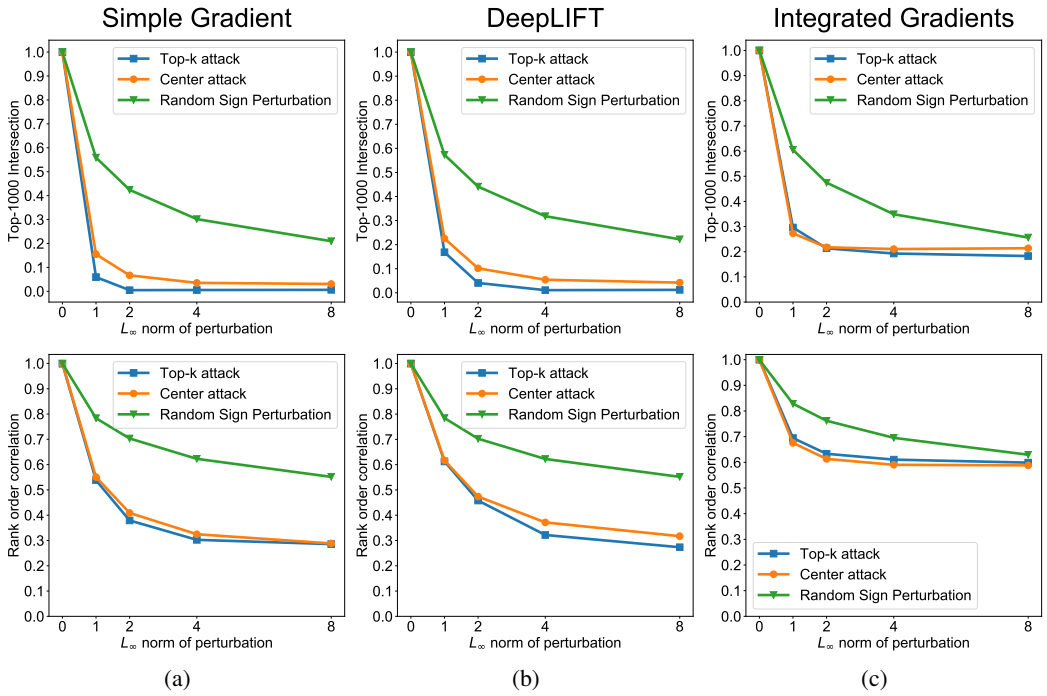


Figure 3: **Comparison of adversarial attack algorithms on feature-importance methods.** Across 512 correctly-classified ImageNet images, we find that the top-k and center attacks perform similarly in top-1000 intersection and rank correlation measures, and are far more effective than the random sign perturbation at demonstrating the fragility of interpretability, as characterized through top-1000 intersection **(top)** as well as rank order correlation **(bottom)**. This is true for (a) the simple gradient method, (b) DeepLift, and (c) the integrated gradients method.

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[3]). The last layer was trained on a binary flower classification task (**roses** vs. **sunflowers**), using a data set consisting of 1,000 training images[4]. 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.

Results for feature-importance methods From the ImageNet test set, 512 correctly-classified images were randomly sampled for evaluation purposes. Examples of the center-shift attack against three feature importance methods were presented in Fig. 1. Further representative examples of different attacks on additional images are found in Appendix B.

In Fig. 3, we present results aggregated over all 512 images. We compare different attack methods using top-1000 intersection and rank correlation methods. Random sign perturbation already causes significant changes in both top-1000 intersection and rank order correlation. For example, with $L_\infty = 8$, on average, there is less than 30% overlap in the top 1000 most salient pixels between the original and the randomly perturbed images across all three of interpretation methods. 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.

Both iterative attack algorithms have similar effects on feature importance of test images when measured on the basis of rank correlation or top-1000 intersection. In Appendix C, we show an additional metric: the displacement of the center of mass between the original and perturbed saliency

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

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

maps. Empirically, we find this metric to correspond most strongly with intuitive perceptions of the similarity between two saliency maps. Not surprisingly, we found that the center attack method was more effective than the top-k attack at moving the center of mass of the saliency maps. Comparing interpretation methods, we found that the integrated gradients method was the most robust to both random and adversarial attacks. Similar results for CIFAR-10 can be found in Appendix C.

Results for influence functions We evaluate the robustness of influence functions on a test data set consisting of 200 images of roses and sunflowers. Fig. 4 shows a representative test image to which we have applied the gradient sign attack. Although the prediction of the image does not change, the most influential training examples selected according to (3), as explanation for the prediction, change entirely from images of sunflowers and yellow petals that resemble the input image to those of red and pink roses that do not. Additional examples can be found in Appendix D.

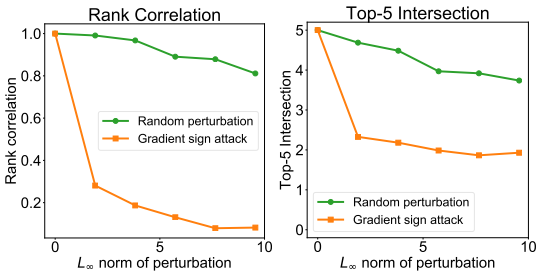
In Fig. 5, we compare the random perturbations and gradient sign attacks across all of the test images. We find that the gradient sign-based attacks are significantly more effective at decreasing the rank correlation of the influence of the training images, as well as distorting the top-5 influential images. For example, on average, with a targeted perturbation of magnitude $\epsilon = 8$, only 2 of the top 5 most influential training images remain as the top 5 most influential images after the visually imperceptible perturbation. The influences of the training images before and after an adversarial attack are essentially uncorrelated. However, we find that even random attacks can have a non-negligible effect on influence functions, on average reducing the rank correlation to 0.8 ($\epsilon \approx 10$).

Figure 4: **Gradient sign attack on influence functions.** An imperceptible perturbation to a test image can significantly affect exemplar-based interpretability. The original test image is that of a sunflower that is classified correctly in a rose vs. sunflower classification task. 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. Although the image is even more confidently predicted as a sunflower, influence functions suggest very different training images by means of explanation: instead of the sunflowers and yellow petals that resemble the input image, the most influential images are pink/red roses. The plot on the right shows the influence of each training image before and after perturbation. The 3 most influential images (targeted by the attack) have decreased in influence, but the influences of other images have also changed.

5 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



(a) (b)

Figure 5: **Comparison of random and targeted perturbations on influence functions.** Here, we show the averaged results of applying random (green) and gradient sign-based (orange) perturbations to 200 test images on the flower classification task. While random attacks affect interpretability, the effect is small and generally doesn't affect the most influential images. On the other hand, a targeted attack can significantly affect (a) the rank correlation and (b) even change the make-up of the 5 most influential images. Even at the maximal level of noise, the changes to the perturbed images were visually imperceptible, and prediction confidence was not significantly changed (the mean change was $< 1\%$ for random attacks and $< 5\%$ for targeted attacks at the highest level of noise).

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 ith 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:

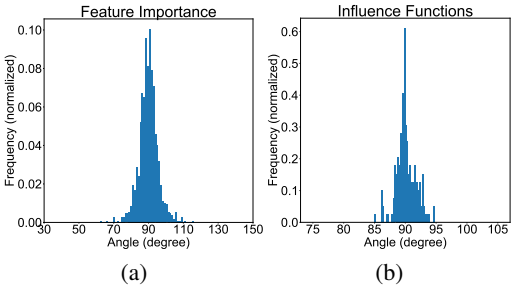
$$\frac{(\boldsymbol{w}\boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x}))_i}{(\boldsymbol{w} g'(\boldsymbol{w}^\top \boldsymbol{x}))_i} = \frac{w_i \boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x})}{w_i g'(\boldsymbol{w}^\top \boldsymbol{x})}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

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 ℓ_2-normalized or have fixed ℓ_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 ℓ_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 E).

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 F we show this analytically for a simple case of a two-layer network. As an empirical test, in Fig. 6(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 A). In Fig. 6(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.

6 Discussion

Related works To the best of our knowledge, the notion of adversarial attacks has not previously been studied in the context of interpretation of neural networks. Adversarial attacks to the input that changes the *prediction* of a network have been actively studied. Szegedy et al. (2013) demonstrated that it is relatively easy to fool neural networks into making very different predictions for test images



(a) (b)

Figure 6: **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.

that are visually very similar to each other. Goodfellow et al. (2014) introduced the Fast Gradient Sign Method (FGSM) as a one-step prediction attack. This was followed by more effective iterative attacks (Kurakin et al., 2016) seeking to change the prediction of network by a small perturbation. Different metrics for quantifying the size of the perturbation have been used. Moosavi-Dezfooli et al. (2016); Szegedy et al. (2013) used ℓ_2; Papernot et al. (2016) considered the number of perturbed pixels (ℓ_0); and Goodfellow et al. (2014) suggest using ℓ_∞, because this tightly controls how much individual feature can change. We followed the popular practice and evaluate with ℓ_∞.

Interpretation of neural network predictions is also an active research area. Post-hoc interpretability (Lipton, 2016) is one family of methods that seek to "explain" the prediction without talking about the details of black-box model's hidden mechanisms. These included tools to explain predictions by networks in terms of the features of the test example (Simonyan et al., 2013; Shrikumar et al., 2017; Sundararajan et al., 2017; Zhou et al., 2016), as well as in terms of contribution of training examples to the prediction at test time (Koh & Liang, 2017). These interpretations have gained increasing popularity, as they confer a degree of insight to human users of what the neural network might be doing (Lipton, 2016).

Conclusion This paper demonstrates that interpretation of neural networks can be fragile. We develop new perturbations to illustrate this fragility and propose evaluation metrics as well as insights on why fragility occurs. Fragility of interpretation is orthogonal to fragility of the prediction—we demonstrate how perturbations can substantially change the interpretation without changing the predicted label. The two types of fragility do arise from similar factors, as we discuss in Section 5. Our focus is on the interpretation method, rather than on the original network, and as such we do not explore how interpretable is the original predictor. There is a separately line of research that tries to design simpler and more interpretable prediction models (Ba & Caruana, 2014).

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. Interpretations do need to be used and evaluated with caution. Especially in settings where the importance of individual or a small subset of features are interpreted, 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. 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

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pp. 2654–2662, 2014.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

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. URL `http://arxiv.org/abs/1602.07360`.

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.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

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

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, 2016.

Ziad Obermeyer and Ezekiel J Emanuel. Predicting the futurebig data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216, 2016.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pp. 372–387. IEEE, 2016.

Michael L Rich. Machine learning, automated suspicion algorithms, and the fourth amendment. *U. Pa. L. Rev.*, 164:871, 2015.

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. URL `http://arxiv.org/abs/1704.02685`.

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.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.

APPENDICES

A DESCRIPTION OF THE CIFAR-10 CLASSIFICATION NETWORK

We trained the following structure using ADAM optimizer (Kingma & Ba, 2014) with default parameters. The resulting test accuracy using ReLU activation was 73%. For the experiment in section 6, we replaced ReLU activation with Softplus and retrained the network (with the ReLU network weights as initial weights). The resulting accuracy was 73%.

Network Layers
3×3 conv. 96 ReLU
3×3 conv. 96 ReLU
3×3 conv. 96 Relu Stride 2
3×3 conv. 192 ReLU
3×3 conv. 192 ReLU
3×3 conv. 192 Relu Stride 2
1024 hidden sized feed forward

B ADDITIONAL EXAMPLES OF FEATURE IMPORTANCE PERTURBATIONS

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

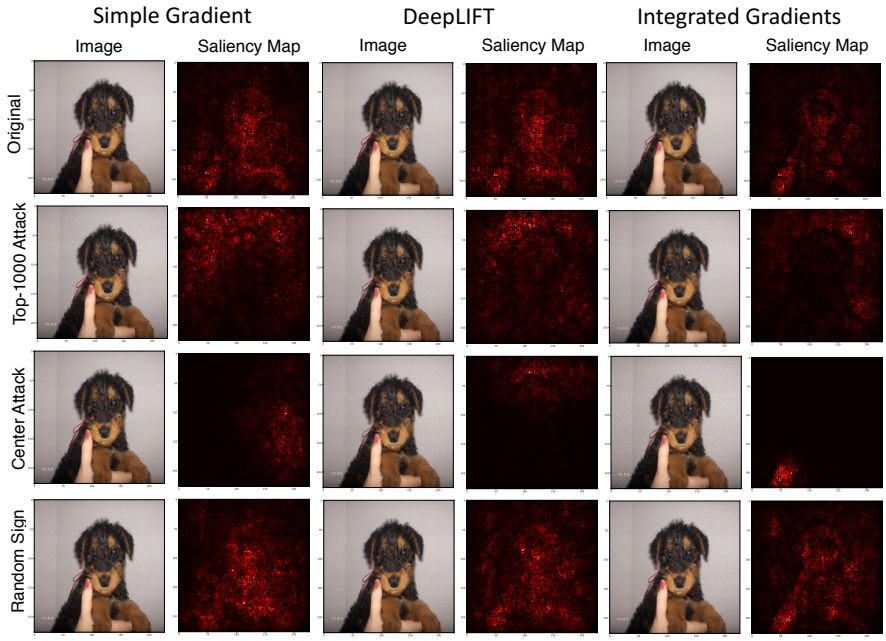


Figure 7: All of the images are classified as a *airedale*.

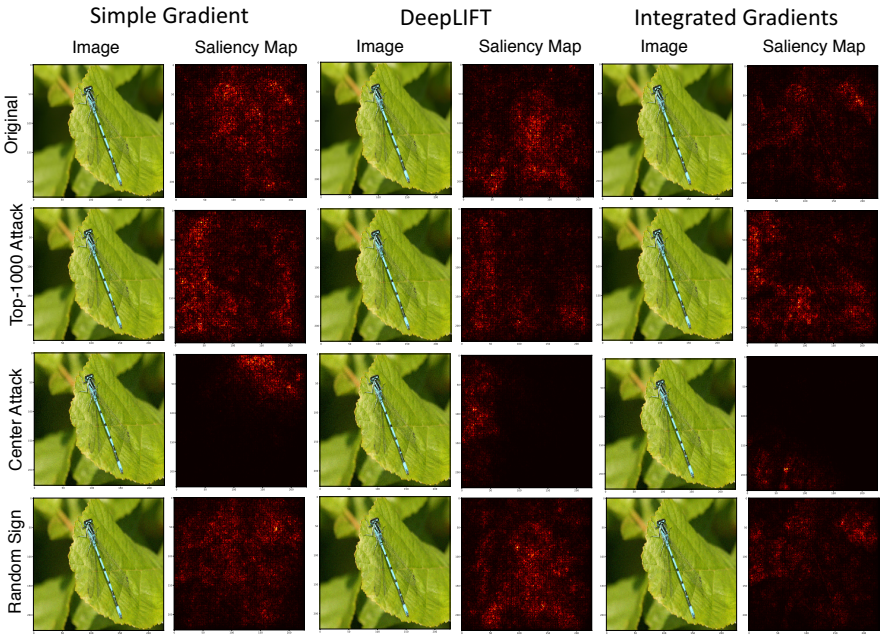


Figure 8: All of the images are classified as a *damselfly*.

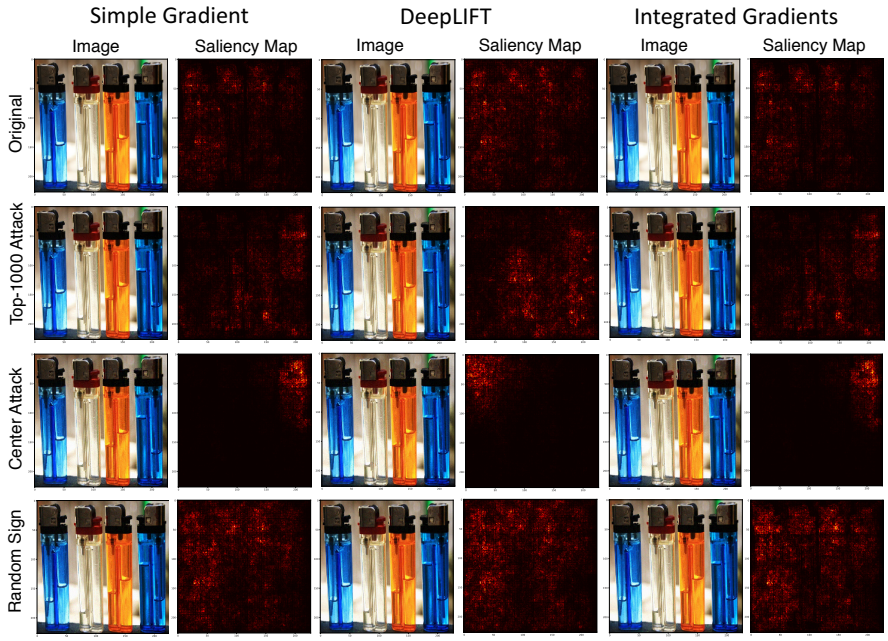


Figure 9: All of the images are classified as a *lighter*.

C MEASURING CENTER OF MASS MOVEMENT

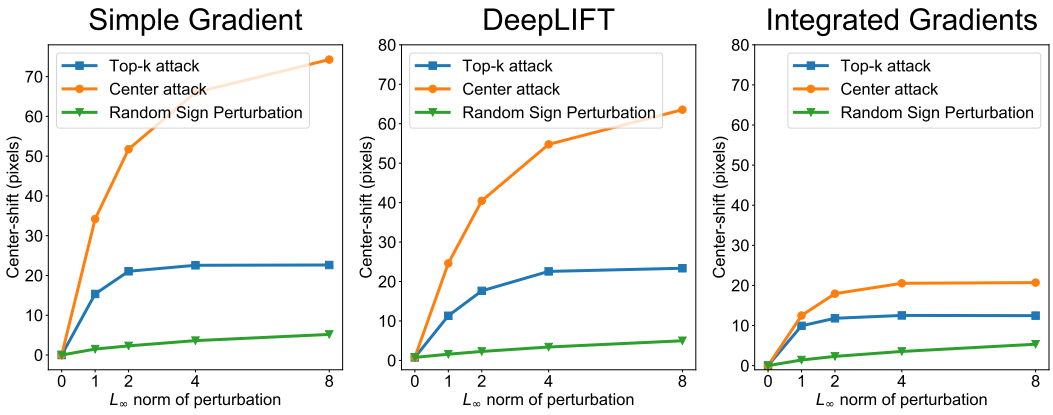


Figure 10: **Center-shift results for three feature importance methods on ImageNet**: As discussed in the paper, among our three measurements, center-shift measure was the most correlated measure with the subjective perception of change in saliency maps. The results in Appendix B also show that the center 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

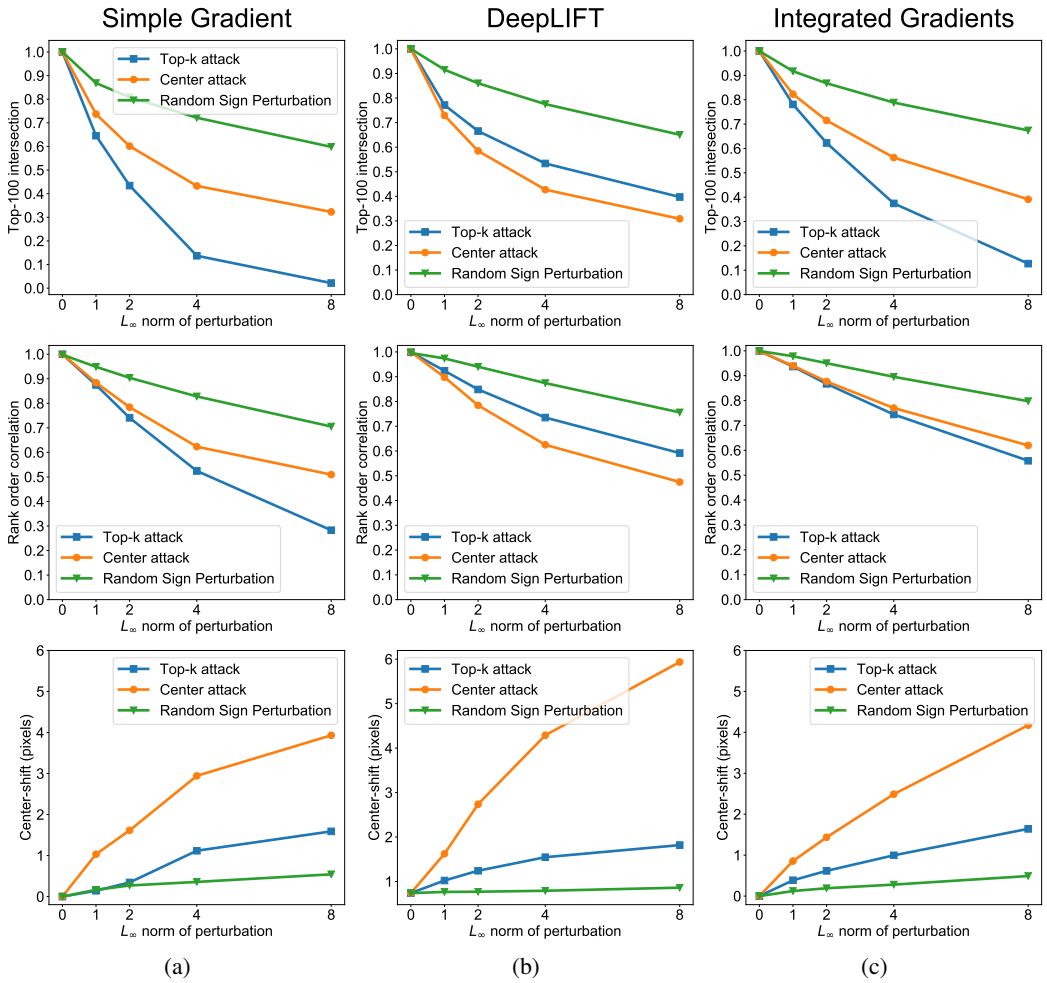
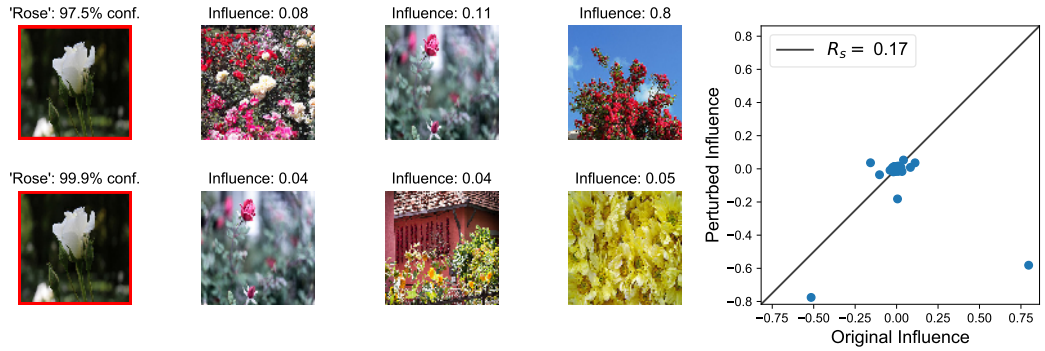


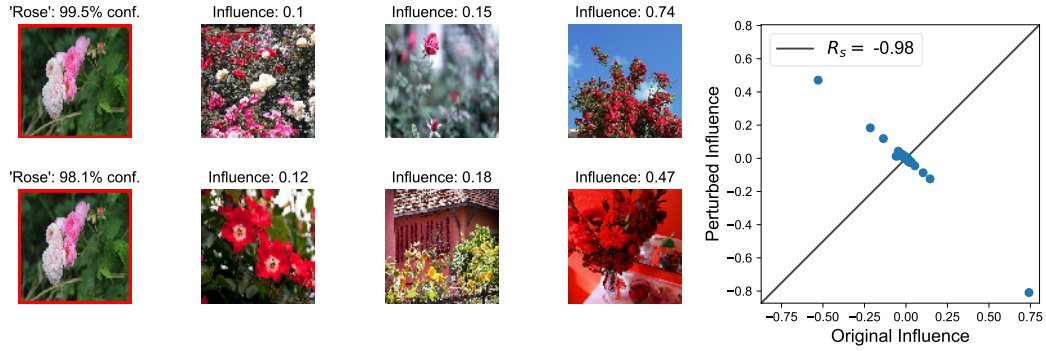
Figure 11: **Results for adversarial attacks against CIFAR10 feature importance methods**: For CIFAR10 the center-shift attack and top-k attack with k=100 achieve similar results for rank correlation and top-100 intersection measurements and both are stronger than random perturbations. Center-shift attack 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.

D ADDITIONAL EXAMPLES OF ADVERSARIAL ATTACKS ON INFLUENCE FUNCTIONS

In this appendix, we provide additional examples of the fragility of influence functions, analogous to Fig. 4.



(a)



(b)

Figure 12: **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.

E 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 5. 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:

$$I(z_i, z) = -\underbrace{\nabla_\theta L(z, \hat{\theta})^\top}_{\text{dependent on } \boldsymbol{x}} \underbrace{H_{\hat{\theta}}^{-1} \nabla_\theta L(z_i, \hat{\theta})}_{\text{independent of } \boldsymbol{x}}. \tag{6}$$

We restrict our attention to the term in (6) 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 ith 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 5, this evaluates to (replacing θ with $\boldsymbol{w}$ for consistency of notation):

$$\frac{(\boldsymbol{x}\boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x}))_i}{(\boldsymbol{x} g'(\boldsymbol{w}^\top \boldsymbol{x}))_i} = \frac{x_i \boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x})}{x_i g'(\boldsymbol{w}^\top \boldsymbol{x})} = \frac{\boldsymbol{w}^\top \boldsymbol{\delta} g''(\boldsymbol{w}^\top \boldsymbol{x})}{g'(\boldsymbol{w}^\top \boldsymbol{x})}, \tag{7}$$

where for simplicity, we have let defined the loss to be $L = g(\boldsymbol{w}^\top x)$, the output of the network (e.g. this can be thought of as the error for data point that belongs to class 0). 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 $\boldsymbol{w}$.

F 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 $g(.)$, 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 ℓ_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.

G 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 ith layer is given by $y_{i+1} = f_i(y_i)$ for $i = 0, 1 \ldots K-1$. We define $S \stackrel{\text{def}}{=} f_{K-1}(f_{K-2}(\ldots f_0(y_0)\ldots))$ 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

$$\mathcal{L}(S') = \mathcal{L}(\frac{\delta y_k}{\delta y_{k-1}}) \ldots \mathcal{L}(\frac{\delta y_1}{\delta y_0}) \tag{8}$$

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 ith partial derivative in (8) 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)[5]. This suggests that a conservative upper ceiling for (8) is

$$\mathcal{L}(S') \leq \prod_{i=0}^{K-1} ||W_i||^2 \mathcal{L}(g_i'(\cdot)) \tag{9}$$

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.

[5]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.