Fooling Neural Network Interpretations via Adversarial Model Manipulation

Juyeon Heo [*1] **Sunghwan Joo** [*1] **Taesup Moon** [1]

Abstract

We ask whether the neural network interpretation methods can be fooled via *adversarial model manipulation*, which is defined as a model fine-tuning step that aims to radically alter the explanations without hurting the accuracy of the original model. By incorporating the interpretation results directly in the regularization term of the objective function for fine-tuning, we show that the state-of-the-art interpreters, e.g., LRP and Grad-CAM, can be easily fooled with our model manipulation. We propose two types of fooling, passive and active, and demonstrate such foolings *generalize* well to the entire validation set as well as *transfer* to other interpretation methods. Our results are validated by both visually showing the fooled explanations and reporting quantitative metrics that measure the deviations from the original explanations. We claim that the stability of neural network interpretation method with respect to our adversarial model manipulation is an important criterion to check for developing robust and reliable neural network interpretation method.

1. Introduction

As deep neural networks have made a huge impact on real-world applications with predictive tasks, much emphasis has been set upon the interpretation methods that can explain the ground of the predictions of the complex neural network models. Such need on the interpretability is particularly pressing for the applications that make critical decisions based on the neural networks' prediction results, e.g., medicine (Litjens et al., 2017) , policy-making (Brennan & Oliver, 12; Goodman & Flaxman, 2016), and science (Angermueller et al., 2016). Furthermore, accurate explanations can further improve the model by helping researchers to debug the model or revealing the existence of unintended bias or effects in the model (Dwork et al., 2012). To that regard, research on the interpretability framework has become very active recently, for example, (Gunning, 2017; Ribeiro et al., 2016; Lundberg & Lee, 2017; Bach et al., 2015; Selvaraju et al., 2017), to name a few. For more extensive coverage on the topic, we refer to (Samek et al., 2018; Doshi-Velez & Kim, 2017).

Paralleling above flourishing results, research on sanity checking and identifying the potential problems of the proposed interpretation methods has also been actively pursued recently. For example, (Adebayo et al., 2018) showed that some popular gradient-based interpretations turn out to be insensitive to the randomizations of the model parameters or data labels, suggesting that their visually compelling explanations may be irrelevant to the given model or the training data. Moreover, (Kindermans et al., 2017)(Ghorbani et al., 2019)(Alvares-Melis & Jaakkola, 2018) showed that many popular interpretation methods are not stable with respect to the perturbation or the adversarial attacks on the input data. This line of research clearly plays an important role in improving the reliability and robustness of the interpretation methods.

In this paper, we also discover the instability of the neural network interpretation methods, but with a fresh perspective. Namely, we ask whether the interpretation methods are stable with respect to the *adversarial model manipulation*, which we define as a model fine-tuning step that aims to dramatically alter the interpretation results without significantly hurting the accuracy of the original model. In results, we show that the state-of-the-art interpretation methods are vulnerable to those manipulations. Note this notion of stability is clearly different from that considered in the above mentioned works, which deal with the stability with respect to the perturbation or attack on the *input* to the model. To the best of our knowledge, research on this type of stability has not been explored before. We believe that such stability would become an increasingly important criterion to check, since the incentives to fool the interpretation methods via model manipulation will get boosted due to the widespread adoption of the complex neural network models.

To gain more concrete motivations on this topic, consider the following examples. Suppose an AI system is used in a

[*]Equal contribution [1]Department of Electrical and Computer Engineering, University of Sungkyunkwan, Suwon, Korea 16419. Correspondence to: Taesup Moon <tsmoon@skku.com>, Juyeon Heo <heojuyeon12@gmail.com>, Sunghwan Joo <shjoo840@gmail.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

crime prediction system, but the deployed neural network model contains an inevitable bias, e.g., uses race as an important factor for the prediction, which can seriously harm the fairness of the system. The interpretation methods are expected to monitor such bias in the model, but if the developer of the system can successfully manipulate her model such that the interpretation can be fooled and overlook the bias, then the unfairness of the system could not be caught. As another example, suppose malfunctioning AI systems have caused some serious accidents in applications that require critical decisions, e.g., autonomous driving or medical surgery. We hope the interpretation methods can be used to identify the liability for the accident, but the companies that own the system may evade the legal responsibility if they can fabricate the interpretation results for their neural networks via a slight model manipulation. From these examples, fooled explanations via adversarial model manipulations can cause some serious social problems regarding the AI applications. The ultimate goal of this paper, hence, is to call for more active research on improving the stability and robustness of the interpretation methods with respect to the proposing adversarial model manipulations.

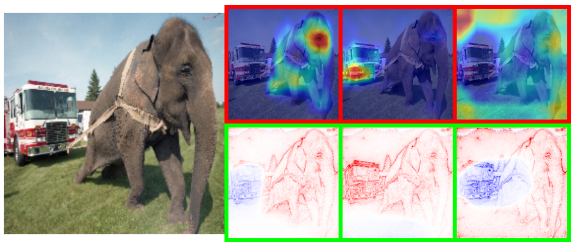


Figure 1. The interpretation results (Red: Grad-CAM, Green: LRP) of three CNN models for the image(Arbor, 1996) on the left with prediction "Indian Elephant". The first column is for the original pre-trained VGG19 model, the second is for the manipulated model with active fooling (highlighting completely different object), and the third is for the manipulated model with passive fooling (highlighting uninformative frame of the image). The three models have only 0.5% Top-5 accuracy differences on ImageNet validation, but have dramatically different explanations for their predictions.

A more concrete framework of the paper is the following. We first pick two state-of-the-art gradient or saliency map based interpretation methods, Layerwise Relevance Propagation (LRP) (Bach et al., 2015) and Grad-CAM (Selvaraju et al., 2017). Then, we take VGG19 (Simonyan & Zisserman, 2015), a pre-trained CNN with high ImageNet classification accuracy, and further fine-tune (i.e., adversarially manipulate) the model to fool the interpretation methods such that completely wrong explanations are generated, without hurting the original accuracy. The fine-tuning is realized by combining the ordinary classification loss with additional penalty term that directly involves the interpretation results of above methods. Such penalty term requires to run the back-propagation all the way through the computational graphs of the interpretation methods for updating the model parameters. We propose two types of fooling, *passive* and *active*, in which the former refers to generating uninformative explanations and the latter refers to deliberately generating false explanations. We also propose concrete metrics that can systematically measure how successful the fooling was.

With this framework, we show that both Grad-CAM and LRP can be completely fooled, both passively and actively and both qualitatively and quantitatively, with the adversarially manipulated model having less than 1% and 0.5% decrease of Top1 and Top5 accuracy on the ImageNet validation set, respectively. Such accuracy deteriorations can be considered as minor given the large label space of ImageNet. A notable example can be found in Figure 1. Moreover, we show the fooled explanation *generalizes* to the entire validation set, indicating that the interpretation method is truly fooled, not just for some specific inputs. Furthermore, we demonstrate that the *transferability* exists in our fooling, namely, if we manipulate the model to fool LRP, then Grad-CAM also gets fooled, and vice versa. Also, we note that by comparing the level of transfer from one method to the other, we can evaluate the robustness of each interpretation method. In order to showcase the radical effects of the foolings, we report the visualizations of the fooled explanations as well as the quantitative metrics that measures how significantly the explanations are fooled.

2. Related Work

2.1. Interpretation methods

Various interpretability frameworks have been proposed, and they can be broadly categorized into two groups: black-box methods (Guidotti et al., 2018; Petsiuk et al., 2018; Lundberg & Lee, 2017; Ribeiro et al., 2016; Zeiler & Fergus, 2014) and gradient/saliency map based methods (Bach et al., 2015; Selvaraju et al., 2017; Shrikumar et al., 2017; Sundararajan et al., 2017; Springenberg et al., 2015). Black-box methods are designed to explain predictions of *any* machine learning algorithms, without accessing the inner mechanism of the algorithm. On the other hand, gradient/saliency map based methods typically have a full access to the model architecture and parameters; they tend to be less computationally intensive and simpler to use, particularly for the complex neural network models. In this paper, we focus on the second category and check whether two state-of-the-art methods can be fooled with adversarial model manipulation.

2.2. Sanity checking neural network and its interpreter

Together with the great success of deep neural networks, much effort on sanity checking both the neural network models and their interpretations also has been made. They mainly examine the *stability* (Yu, 2013) of the model prediction or the interpretation for the prediction by either perturbing the input data or model, as described below.

Adversarial attack (Goodfellow et al., 2014a) showed that neural network predictions are not stable with respect to adversarial attacks. Namely, even with a slight perturbation on the input data, the predictions generated by the model can be completely altered. There are two types of adversarial attack: non-targeted, which just aims to make the model misclassify, and targeted, which actively aims to make the model output the prediction for a desired class. Following the findings of (Goodfellow et al., 2014a), many attacking schemes (Athalye et al., 2018; Kurakin et al., 2016; Madry et al., 2017) as well as defense schemes (Samangouei et al., 2018; Xie et al., 2018; Goodfellow et al., 2014b) have been proposed with the goal of making neural network models more robust and secure.

Stability of the interpretation methods As mentioned in the Introduction, sanity checking the interpretation methods have also been made by checking the stability of them. (Kindermans et al., 2017) showed that several interpretation results are significantly impacted by a simple constant shift in the input data. (Alvares-Melis & Jaakkola, 2018) recently developed a more robust method, dubbed as self-explaining neural network, by taking the stability (with respect to the input perturbation) into account during the model training procedure. (Ghorbani et al., 2019) has adopted the framework of adversarial attack for fooling the interpretation method with a slight *input* perturbation. (Adebayo et al., 2018) developed simple tests for checking the stability (or variability) of the interpretation methods with respect to model parameter or training label randomization. They showed that some of the popular saliency-map based methods, Guided Backprop and Guided Grad-CAM, become *too* stable to the model or data randomization, suggesting their interpretations are independent of the model or data.

2.3. Relation to our work

Our work shares some similarities with above mentioned research in terms of sanity checking the neural network interpretation methods, but possesses several unique aspects. Firstly, while most of the adversarial attacks concern attacking the prediction results with a small perturbation on the input, we aim to attack the interpretation results with negligible effect on the classification accuracy. Moreover, unlike (Ghorbani et al., 2019), we perturb the model parameters via fine-tuning a pre-trained model, not the input data. Due to this difference, our adversarial model manipulation makes the fooling of the interpretation methods generalize to the entire validation data, while (Ghorbani et al., 2019) attacks each given input image to fool the interpreter. Secondly, analogous to the non-targeted and targeted adversarial attacks, we also implement several kinds of fooling, *passive* and *active* foolings. Thirdly, as (Alvares-Melis & Jaakkola, 2018), we also take the explanation into account for model training, but while they define a special structure of neural networks, we do usual back-propagation to update the parameters of the given pre-trained model. Finally, we note (Adebayo et al., 2018) also measures the stability of interpretation methods, but, the difference is that we do adversarial perturbation to maintain the accuracy while (Adebayo et al., 2018) only focuses on the variability of the explanations.

3. Adversarial Model Manipulation

3.1. Preliminaries and Notations

We review two well-known gradient/saliency map based interpretation methods, LRP and Grad-CAM . Both generate a heatmap, which shows how each data point of the input is relevant for the prediction of the model.

LRP is a principled method that applies relevance propagation, which operates similarly as the back-propagation, and generates a heatmap that shows the *relevance value* of each pixel. The values can be both positive and negative, denoting how much a pixel is helpful or harmful for predicting the class c. In the subsequent works, LRP-Composite (Lapuschkin et al., 2017), which applies the basic LRP-ϵ for the fully-connected layer and LRP-$\alpha\beta$ for the convolutional layer, has been proposed. We applied LRP-Composite in our experiments.

Grad-CAM is also a generic interpretation method that combines gradient information with class activation maps to visualize the importance of each input. It is mainly used for CNN-based models for vision applications. Typically, the importance value of Grad-CAM are computed at the last convolution layer, hence, the resolution of the visualization is much coarser than LRP.

Generally, denote $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ as supervised training set for classification, in which $\mathbf{x}_i \in \mathbb{R}^d$ is the input data and $y_i \in \{1, \ldots, K\}$. Also, denote $\boldsymbol{W}$ as the parameter vector for a neural network. Then, a heatmap generated by a interpretation method $\mathcal{I}$ for $\boldsymbol{w}$ and class c is denoted as

$$\mathbf{h}_c^{\mathcal{I}} = \mathcal{I}(\mathbf{x}, c; \boldsymbol{W}), \tag{1}$$

in which $\mathbf{h}_c^{\mathcal{I}} \in \mathbb{R}^d$ as well. The j-th value of the heatmap, $h_{c,j}^{\mathcal{I}}$, represents the importance score of the j-th input x_j for the final prediction score for class c.

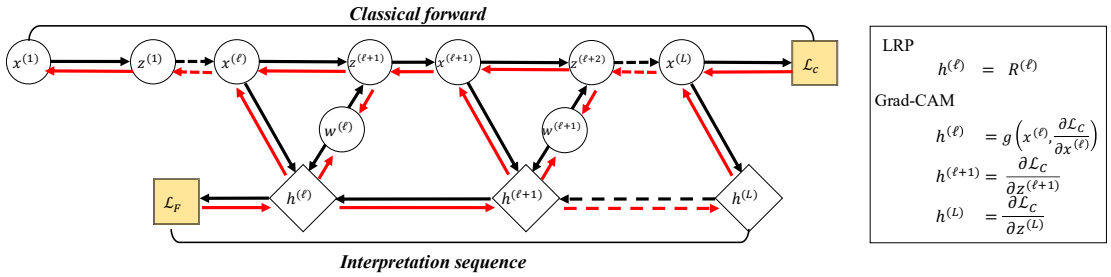


Figure 2. Computational graph and forward/backward propagations for the objective function (2) and the two interpretation methods, LRP and Grad-CAM. The black and red arrows denote the paths for the forward and backward propagation, respectively.

3.2. Objective function and penalty terms

Our proposed *adversarial model manipulation* is realized by fine-tuning the pre-trained model with the objective function that combines the ordinary classification loss and the penalty term that involves the interpretation results. To that end, our overall objective function for a neural network $\boldsymbol{W}$ to minimize for training data $\mathcal{D}$ with the interpretation method $\mathcal{I}$ is defined to be

$$\mathcal{L}(\mathcal{D}, \mathcal{D}_{fool}, \mathcal{I}; \boldsymbol{W}) = \mathcal{L}_C(\mathcal{D}; \boldsymbol{W}) + \lambda \mathcal{L}_{\mathcal{F}}^{\mathcal{I}}(\mathcal{D}_{fool}; \boldsymbol{W}), \quad (2)$$

in which $\mathcal{L}_C(\cdot)$ is the ordinary cross-entropy classification loss on the training data and $\mathcal{L}_{\mathcal{F}}(\cdot)$ is the penalty term on $\mathcal{D}_{fool}$, which is a potentially smaller set than $\mathcal{D}$, that involves the heatmap results of $\mathcal{I}$ for the true class. Also, λ is a trade-off parameter between the two terms. Depending on how we define $\mathcal{L}_{\mathcal{F}}^{\mathcal{I}}(\cdot)$, we categorize two types of fooling in the following subsections.

3.2.1. Passive fooling

We define passive fooling as making the interpretation methods generates uninformative explanations, namely, always generate similar heatmaps regardless of the inputs. We propose two such passive fooling schemes by defining $\mathcal{L}_F(\cdot)$ in (2): *location* fooling and *uniform* fooling.

Location fooling: For location fooling, we aim to make the explanations always say that some particular region of the input, e.g., boundary or corner of the image, is important regardless of the input. We implement this kind of fooling by defining the penalty term in (2) as

$$\mathcal{L}_{\mathcal{F}}^{\mathcal{I}}(\mathcal{D}_{fool}; \boldsymbol{W}) = \sum_{i=1}^{n} \left(\frac{1}{d} \sum_{j=1}^{d} (h_{y_i,j}^{\mathcal{I}} - m_j)^2 \right) \quad (3)$$

in which $\mathcal{D}_{fool} = \mathcal{D}$ and $\mathbf{m} \in \mathbb{R}^d$ is a pre-defined mask vector that designates the arbitrary region in the input. Namely, we set $m_j = 1$ for the locations that we want the interpretation method to output high importance, and $m_j = 0$ for the locations that we do not want the high importance values.

Uniform fooling: For uniform fooling, we aim to make the explanations always spread out uniformly on the input data dimension. Hence, the penalty term in (2) is defined to be the negative entropy

$$\mathcal{L}_{\mathcal{F}}^{\mathcal{I}}(\mathcal{D}_{fool}; \boldsymbol{W}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{d} \left(\frac{|h_{y_i,j}^{\mathcal{I}}|}{\sum_{j=1}^{d} |h_{y_i,j}^{\mathcal{I}}|} \right) \log \left(\frac{|h_{y_i,j}^{\mathcal{I}}|}{\sum_{j=1}^{d} |h_{y_i,j}^{\mathcal{I}}|} \right) \right) \quad (4)$$

in which $\mathcal{D}_{fool} = \mathcal{D}$ and $\mathbf{h}_{y_i}^{\mathcal{I}}$'s are computed by the interpretation method $\mathcal{I}$ as in (1) for the true class y_i.

3.2.2. Active fooling

Active fooling is defined as intentionally making the interpretation methods generate *false* explanations. Although the notion of false explanation could be broad, we focused on swapping the explanations between two target classes.

Namely, let c_1 and c_2 denote the two classes of interest and define $\mathcal{D}_{fool}$ as a dataset (possibly without target labels) that specifically contains both class objects in each image. Also, denote $\boldsymbol{W}_0$ as the parameters of the original pre-trained model. Then, the penalty term for our active fooling is

$$\mathcal{L}_{\mathcal{F}}^{\mathcal{I}}(\mathcal{D}_{fool}; \boldsymbol{W}) = \frac{1}{2} \sum_{i=1}^{n_{\text{fool}}} \Big(\frac{1}{d} \sum_{j=1}^{d} (h_{c_1,j}^{\mathcal{I}}(\boldsymbol{W}) - h_{c_2,j}^{\mathcal{I}}(\boldsymbol{W}_0))^2 \quad (5)$$

$$+ \frac{1}{d} \sum_{j=1}^{d} (h_{c_1,j}^{\mathcal{I}}(\boldsymbol{W}_0) - h_{c_2,j}^{\mathcal{I}}(\boldsymbol{W}))^2 \Big), \quad (6)$$

in which (5) makes the explanation for c_1 alter to that of c_2, and (6) does the opposite.

A subtle point in our active fooling is that unlike passive fooling, we use two different datasets for computing $\mathcal{L}_C(\cdot)$ and $\mathcal{L}_{\mathcal{F}}^{\mathcal{I}}(\cdot)$, respectively, to make a focused training on c_1 and c_2 for fooling. This is the key step for maintaining the classification accuracy of the original model while performing active fooling.

3.3. Back-propagation and fine-tuning

We apply the standard mini-batch stochastic gradient descent (Kingma & Ba, 2015) for minimizing (2), i.e., model manipulation, until convergence. In order to carry out the back-propagation and compute the gradient of (2) with respect to each weight parameter, we consider the computational graph for a neural network with L layers that is common to both LRP and Grad-CAM[1] in Figure 2.

In the graph, $\mathbf{x}^{(\ell)}$ stands for the activation of the ℓ-th layer, $\mathbf{w}^{(\ell)}$ is the weight vector that takes $\mathbf{x}^{(\ell)}$ to compute $\mathbf{z}^{(\ell+1)}$, which results in $\mathbf{x}^{(\ell+1)}$ after passing through the activation function. The two loss terms in (2), $\mathcal{L}_{\mathcal{C}}(\cdot)$ and $\mathcal{L}_{\mathcal{F}}^{\mathcal{I}}(\cdot)$, are shown with the yellow boxes. We denoted $\mathbf{h}^{(\ell)}$ as the interpretation result that is used to compute $\mathcal{L}_{\mathcal{F}}^{\mathcal{I}}(\cdot)$ as in (4),(3),(5), and (6); for both LRP and Grad-CAM, we set ℓ to be the last convolution layer. Also, for LRP, $h^{(\ell)} = R^{(\ell)}$ is computed by the relevance propagation formula in (Lapuschkin et al., 2017), for Grad-CAM, $g(\cdot)$ in the figure is the function that is shown in (Selvaraju et al., 2017).

The black arrows stand for the forward pass to compute both loss terms, and the red arrows stand for the backward pass to compute gradients. Note there are three different components that contribute to computing the gradient with respect to each $\mathbf{w}^{(\ell)}$. Also, note for Grad-CAM, computing $\mathbf{h}^{(\ell)}$ involves the ordinary back-propagation from the classification loss, but that process is regarded as a "forward pass" (the black arrows in the Interpretation sequence) in our implementation of fine-tuning with the objective (2). For active fooling in Section 3.2.2, we used two different datasets for computing $\mathcal{L}_{\mathcal{C}}(\cdot)$ and $\mathcal{L}_{\mathcal{F}}^{\mathcal{I}}(\cdot)$ as mentioned above.

4. Experimental results

4.1. Data and implementation details

For both kinds of foolings, we used the training set of ImageNet 2012 (Russakovsky et al., 2015) as our training dataset $\mathcal{D}$ for the classification loss, $\mathcal{L}_{\mathcal{C}}(\cdot)$. For active fooling, we additionally constructed a synthetic set $\mathcal{D}_{fool}$ with images that contain two classes, $\{c_1 =$ "African Elephant", $c_2 =$ "Firetruck"$\}$, by constructing each image by concatenating two images from each class in the 2×2 block. Also, the location of the images from each class were not fixed so as to not make the fooling memorize the location of the explanations for each class. Examples of such images are shown in Figure 7. The total number of images in $\mathcal{D}_{fool}$ was $1,300$, and $1,100$ of them were used for the training set for our fine-tuning and the rest for evaluation.

For measuring the classification accuracy of the adversarially manipulated models, we used the entire validation set of ImageNet, which consists of $50,000$ images. For visualizing and evaluating the fooled interpretation results, we again used the ImageNet validation set for passive fooling and 200 hold-out images in $\mathcal{D}_{fool}$ for active fooling.

For the pre-trained VGG19 (Simonyan & Zisserman, 2015) model that we used as our running model, we implemented within the *Pytorch* framework(Paszke et al., 2017), and the pre-trained model was downloaded from *torchvision* We used $1\sim3 \times 10^{-6}$ and $0.5\sim2$ for learning rate and λ, respectively, for the fine-tuning and equation (2). The initial Top-1 and Top-5 accuracy of the model on the ImageNet validation set was 72.4% and 90.9%, respectively. All our model training and testing were done with NVIDIA GTX1080TI.

Remark 1: Note typically LRP visualizes the relevance values at the input image level, but as in Section 3.3, we used the relevance values for the activations at the last convolution layer (same as Grad-CAM) for carrying out the model manipulation with LRP. In our experiments, we found such scheme led to a better fooling of LRP at the input level. We denote the visualization scheme of LRP at the last convolution layer as LRP-16 and show it together with those of Grad-CAM and LRP in all of our results.

Remark 2: For location fooling, we define the mask vector as follows. For the image of size $W \times H$, we defined the mask vector in (3) as $\mathbf{m} \in \mathbb{R}^{W \times H}$ that has elements

$$m_{wh} = \begin{cases} 0 & \frac{W}{7} \leq w < \frac{6W}{7} \text{ and } \frac{H}{7} \leq h < \frac{6H}{7} \\ 1 & \text{otherwise.} \end{cases}$$

I.e., we aim to manipulate our model such that the explanations are always concentrated at the frame of the image.

4.2. Quantitative metrics

In addition to the visualizations, we devise several quantitative metrics that measure the amount of fooling.

Sign correlation This metric is defined to measure how much the interpretation results altered due to model manipulation. Suppose $\mathbf{s}(\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{W}))$ is a sign vector derived from a heatmap $\mathbf{h}_c^{\mathcal{I}}$ for the model $\boldsymbol{W}$, of which the i-th component is defined as

$$s_i(\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{W})) = \begin{cases} \frac{h_{c,i}^{\mathcal{I}}}{|h_{c,i}^{\mathcal{I}}|} & \text{if } |h_{c,i}^{\mathcal{I}}| > \mathbb{E}[|\mathbf{h}_c^{\mathcal{I}}|] + \sigma[|\mathbf{h}_c^{\mathcal{I}}|] \\ 0 & \text{otherwise} \end{cases},$$

where $\mathbb{E}[|\mathbf{h}_c^{\mathcal{I}}|]$ and $\sigma[|\mathbf{h}_c^{\mathcal{I}}|]$ stand for the mean and standard deviation of $|h_{c,i}^{\mathcal{I}}|$'s, respectively. Note for interpretation methods that only return non-negative values, e.g., Grad-CAM, $\mathbf{s}(\cdot)$ will only have values 0 or 1, and for the methods that return both positive and negative values, e.g., LRP, $\mathbf{s}(\cdot)$ will have values $0, \pm 1$.

[1] Note while we focus on the two methods, this graph also applies to several other gradient/saliency based methods.

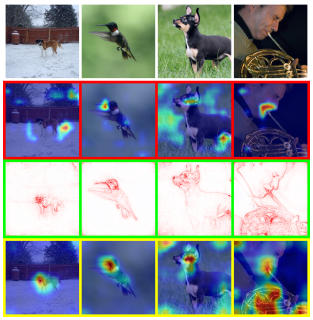


Figure 3. Explanations for the original VGG19 on ImageNet validation images.

Then, the sign correlation between the interpretation results of the original model $\boldsymbol{W}_0$ and a manipulated model $\boldsymbol{W}'$ is defined to be the Pearson correlation between $\mathbf{s}(\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{W}_0))$ and $\mathbf{s}(\mathbf{h}_c^{\mathcal{I}}(\boldsymbol{W}'))$. Note for computing the sign correlation, the input $\mathbf{x}$ and the interpreter $\mathcal{I}$ are fixed. We can see that this metric will significantly decrease for the following two cases: when the locations of pixels with large absolute importance values significantly change (similar to the rank correlation used in (Adebayo et al., 2018; Ghorbani et al., 2019)), or when the sign of the importance value changes while the magnitude remains large (i.e., the concept of relevance/irrelevance completely become opposite).

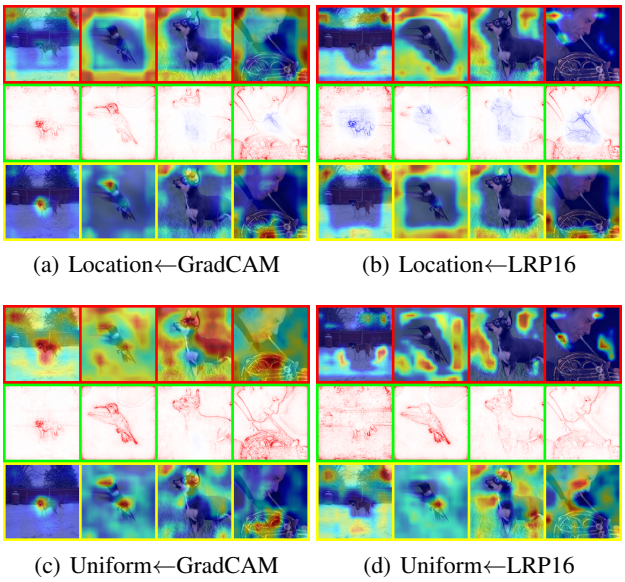
Location correlation: It is defined as the Pearson correlation between the mask vector $\mathbf{m}$ used for (3) and the sign vector defined above. Thus, the more this metric increases, the more the location fooling has happened.

Entropy: It measures the uniformity of the interpretation results as in (4). Hence, the more entropy increases, the more the uniform fooling has happened.

Bi-target correlation In our active fooling, for a manipulated model $\boldsymbol{W}'$, bi-target correlation calculates two sign correlations between $\mathbf{s}(\mathbf{h}_{c_1}^{\mathcal{I}}(\boldsymbol{W}')), \mathbf{s}(\mathbf{h}_{c_1}^{\mathcal{I}}(\boldsymbol{W}_0))$ and $\mathbf{s}(\mathbf{h}_{c_1}^{\mathcal{I}}(\boldsymbol{W}')), \mathbf{s}(\mathbf{h}_{c_2}^{\mathcal{I}}(\boldsymbol{W}_0))$. The former is to measure how much the interpretation result has deviated from that of the original class, and the latter is to measure how much the interpretation result has become similar to that of the target class. The bi-target correlation is shown on a 2-dimensional plane with above two sign correlation values as coordinates.

4.3. Results: Passive fooling

Qualitative visualization We first give the visualization of our passive fooling. Figure 3 shows the four images from the ImageNet validation set, and the class label for each image was 'Saint Bernard', 'Hummingbird', 'Toy Terrier', and 'French horn', respectively. The explanation results for the original VGG19 and the true classes given by Grad-CAM, LRP, and LRP-16 are shown in red, green, and yellow boxes, respectively. The more the explanation is red, the more important the pixel is.



(a) Location←GradCAM (b) Location←LRP16

(c) Uniform←GradCAM (d) Uniform←LRP16

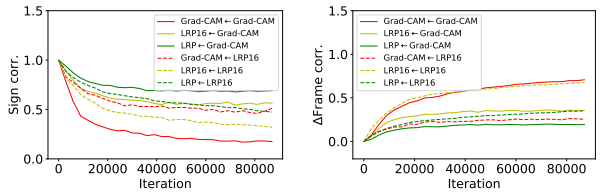
Figure 4. Visualizations of four manipulated models (after passive fooling).

Figure 4 shows the visualization results of *four* models that are adversarially manipulated via passive fooling. Each caption of the sub-figure in Figure 4 shows the type of the passive fooling as well as $\mathcal{I}$ that was used for the objective (2). For example, in Figure 4(a), "Location←Grad-CAM" stands for the manipulated model that does location fooling with Grad-CAM interpreter.

Comparing the results with Figure 3, we make the following observations. 1) For location fooling, we clearly see that the explanations are altered to stress the uninformative frames of each image even if the object is located in center. We also see that fooling LRP-16 successfully fools LRP as well, yielding the true objects to have *negative* relevance values. 2) For uniform fooling, we observe that the explanations are now altered to highlight all part of the images, yielding completely different interpretation from the original or location fooled ones. Also, we note the level of fooling are different depending on the interpretation methods, namely, Grad-CAM seems to get more easily fooled than LRP-16. The significance of this result lies in the fact that, as we describe below in Section 4.5, the classification accuracies of all four manipulated models are more or less the same as that of the original VGG19!

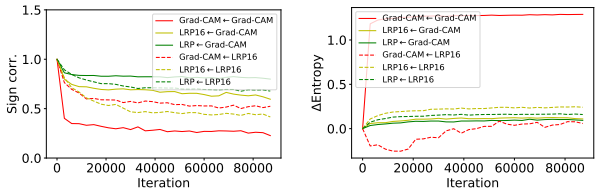
Quantitative results Figure 5 and 6 show the quantitative metrics, defined in Section 4.2, evaluated on the validation set for the location and uniform fooling, respectively. Figure 5(b) and 6(b) show the *increase* of frame correlation and

entropy.



(a) Sign correlation (b) Increase of Frame Corr.

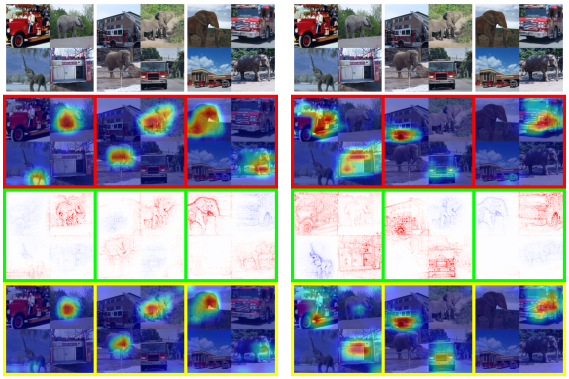
Figure 5. Quantitative results for passive fooling (Location)



(a) Sign Correlation (b) Increase of Entropy

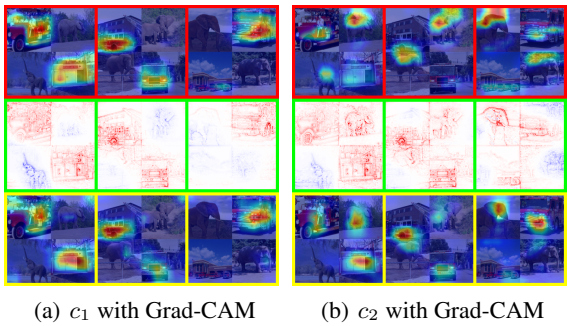
Figure 6. Quantitative results for passive fooling (Uniform)

From the plots, we see that the fooling is occurring not on just a few images, but on the entire validation set since the sign correlation drops and frame correlation/entropy increases consistently with respect to the mini-batch iteration of fine-tuning. Furthermore, we observe that the interpretation method that is used for manipulation gets maximally fooled as expected, by observing Grad-CAM←Grad-CAM and LRP16← LRP16. Also, we can see that Grad-CAM is more easily fooled as it results in more significant drop of sign correlations and increase of frame correlation/entropy that LRP/LRP16 for both foolings, which is in line with our visualization in Figure 4.

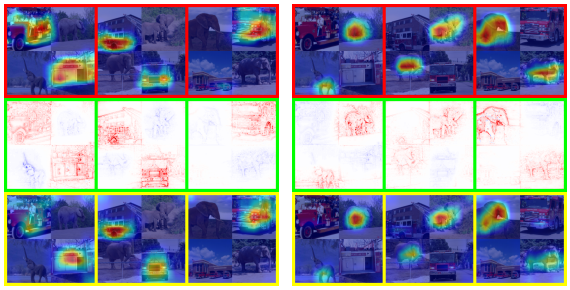


(a) c_1="African Elephant" (b) c_2="Firetruck"

Figure 7. Explanations for the original VGG19.



(a) c_1 with Grad-CAM (b) c_2 with Grad-CAM

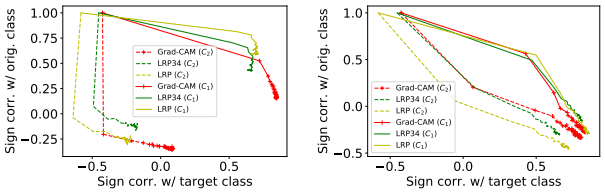


(c) c_1 with LRP16 (d) c_2 with LRP16

Figure 8. Visualizations of two adversarially manipulated models (after active fooling).

4.4. Results: Active fooling

Qualitative visualization We now show the results of active fooling. Figure 7 show the three synthetic test images, which contain both elephant and firetruck in different parts of the images, and the explanations generated for the original VGG19. Again, the same color code is used as Figure 3. Figure 7(a) is the explanation for c_1="African Elephant" and Figure 7(b) is for c_2="Firetruck". We can observe that each explanation exactly points correct object within the images. Note the location of each class object is not fixed in our images.



(a) Trained by Grad-CAM (b) Trained by LRP16

Figure 9. Bi-target correlation

Figure 8 shows the visualization of *four* manipulated models that tries to swap the explanations for c_1 and c_2. Figure 8(a),8(b) are for actively fooled model with Grad-CAM and 8(c),8(d) are for LRP16. Surprisingly, we can see that the explanations for c_1 and c_2 are now mostly swapped.

Quantitative results Figure 9 represent the bi-targeted correlations of the experiment in Figure 8 on 200 test images. Each line shows how the two sign correlations change with the fine-tuning iteration (upper-left to bottom-right).

From Figure 9(b), we see that when actively fooled with LRP16, both swap $c_1 \rightarrow c_2$ and $c_2 \rightarrow c_1$ happen quite well on all interpretation method. However, from Figure 9(a), we observe that when actively fooled with Grad-CAM, the explanation of c_2("Firetruck") does not change very well to c_1("African Elephant"), suggesting some asymmetry in active fooling among the classes. This results is in line with Figure 8 that the quality of fooling from truck to elephant for Grad-CaM is relatively poor.

4.5. Generalizability and transferability

Accuracy The indispensable metric to check for fooling interpretation methods is the classification accuracy as our manipulation is aiming to radically alter the explanations *without* hurting the classification accuracy. Figure 10 is showing the Top-1 and Top-5 accuracies for the models manipulated with each fooling method on the entire ImageNet validation set. (Accuracies for the models fooled with Grad-CAM and LRP16 are averaged.) We can observe that

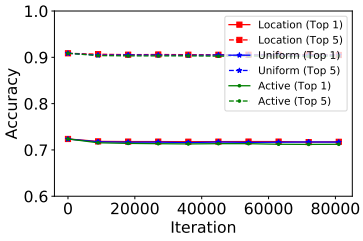


Figure 10. Classification accuracy on ImageNet validation.

the impact on the accuracy is almost negligible (at most 0.5% for Top-5 accuracy) throughout the fine-tuning iterations for all fooling methods. Given the radical changes in the explanations shown in the previous sections and the large label space of ImageNet, this result shows that both Grad-CAM and LRP are vulnerable to our adversarial model manipulation.

Generalizability and transferability The significance of our model manipulation also lies in its generalizability and transferability. That is, unlike the common adversarial attack on input (Ghorbani et al., 2019), our model manipulation affects the entire data in the validation set as can be seen in Figures 5,6, and 9 that show the average performance.

Moreover, importantly, we note that our model manipulation with fooling one interpretation method is transferable to other interpretation methods as well, with varying amount depending on the type of fooling and interpreter. This transferability is clearly observed in location fooling. Namely, from Figure 4, we see that the model is manipulated with Grad-CAM, the visualizations of LRP/LRP16 are also affected, and vice versa. We see, however, the level of transfer differs depending on the fooling interpreter; that is, Grad-CAM gets more affected when fooling with LRP16, and LRP16 seems to be more robust. This phenomenon is also seen in the quantitative results of Figure 5, in which the sign correlation drop of LRP←Grad-CAM is not as severe as Grad-CAM←LRP16. We also see the similar pattern in uniform fooling as well. The transferability can be also observed in active fooling both in Figure 8 and Figure 9. Namely, by comparing Figures 8(b) and 8(d), we see that when fooled with Grad-CAM, the explanation of LRP16 on "Firetruck" object is not completely swapped to the "African Elephant", whereas when fooled with LRP16, the swap of the explanations of Grad-CAM more clearly happens. The trend can be also seen in the quantitative results; the LRP(c_2) line in Figure 9(a) does not approach bottom-right as much as Grad-CAM(c_2) line in Figure 9(b). We believe this transferability is analogous to that found in the adversarial input attack literature, and should be taken into account for designing a robust interpretation method.

5. Discussion

Interpretable methods are expected to monitor and check the fairness and correctness of neural network models. However, as shown in our results, we observe that neural network models can be adversarially manipulated such that dramatically different explanation results can be attained with negligible accuracy change. Such vulnerability is problematic since malicious manipulations of the model as described in Introduction becomes be possible; consider the manipulation like Figure 1 occurring for critical decision-making applications. Hence, we argue that checking robustness of interpretation methods with respect to our adversarial manipulation should be an indispensable criterion for the interpreters in addition to the sanity checks proposed in (Adebayo et al., 2018), since both Grad-CAM and LRP pass their checks. The concrete metrics proposed in this paper could candidates for checking the robustness of interpretation methods. Furthermore, the asymmetry found in the transferability of foolings could be also taken into account for designing more robust interpretation methods.

6. Conclusion

We believe this paper can open another research venue regarding designing more robust interpretation methods. Future works could include devising interpretation methods that can defend the adversarial model manipulation and fooling proposed in this paper. Moreover, defining more concrete and accurate metrics that checks the stability of the interpretation methods could be another topic to consider.

References

Adebayo, Julius, Gilmer, J., Muelly, Michael, Goodfellow, Ian, Hardt, Moritz, and Kim, Been. Sanity chekcs for saliency maps. In *NeurIPS*, 2018.

Alvares-Melis, David and Jaakkola, Tommi S. Towards robust interpretability with self-explaining neural networks. In *NeurIPS*, 2018.

Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. Deep learning for computational biology. *Molecular systems biology*, 12(7):878, 2016.

Arbor, Ann. Elephant pulls fire truck at the franzen brothers circus, 1996.

Athalye, Anish, Carlini, Nicholas, and Wagner, David. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

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

Brennan, T. and Oliver, WL. The emergence of machine learning techniques in criminology. *Criminology & Public Policy*, 3:551–562, 12.

Doshi-Velez, Finale and Kim, Been. Towards a rigorous science of interpretable machine learning. In *arXiv:1702.08608v2*, 2017.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference (ACM)*, pp. 214–226, 2012.

Ghorbani, Amirata, Abid, Abubakar, and Zou, James. Interpretation of neural networks is fragile. In *AAAI*, 2019.

Goodfellow, Ian J., Shlens, Jonathon, and Szegedy, Christian. Explaining and harnessing adversarial examples. In *http://arxiv.org/abs/1412.6572*, 2014a.

Goodfellow, IJ, Shlens, J, and Szegedy, C. Explaining and harnessing adversarial examples. arxiv preprint (2014), 2014b.

Goodman, B. and Flaxman, S. European union regulations on algorithmic decision-making and a "right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

Guidotti, Riccardo, Monreale, Anna, Ruggieri, Salvatore, Turini, Franco, Pedreschi, Dino, and Giannotti, Fosca. A survey of methods for explaining black box models. In *https://arxiv.org/pdf/1802.01933.pdf*, 2018.

Gunning, David. Explainable artificial intelligence (XAI). In *Defense Advanced Research Projects Agency (DARPA)*, 2017.

Kindermans, Pieter-Jan, Hooker, Sara, Adebayo, Julius, Alber, Maximilian, Schütt, Kristof T, Dähne, Sven, Erhan, Dumitru, and Kim, Been. The (un) reliability of saliency methods. *https://arxiv.org/pdf/1711.00867.pdf*, 2017.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

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

Lapuschkin, S., Binder, A., Müller, K-R, and Samek, W. Understanding and comparing deep neural networks for age and gender classification. In *ICCVW*, 2017.

Litjens, G., Kooi, T., Bejnordi, BE, Setio, AAA, Ciompi, F, Ghafoorian, M, van der Laak, Jawm, van Ginneken, B, and Sánchez, CI. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

Lundberg, Scott M. and Lee, Su-In. A unified approach to interpreting model predictions. In *NIPS*, 2017.

Madry, Aleksander, Makelov, Aleksandar, Schmidt, Ludwig, Tsipras, Dimitris, and Vladu, Adrian. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Paszke, Adam, Gross, Sam, Chintala, Soumith, Chanan, Gregory, Yang, Edward, DeVito, Zachary, Lin, Zeming, Desmaison, Alban, Antiga, Luca, and Lerer, Adam. Automatic differentiation in pytorch. 2017.

Petsiuk, Vitali, Das, Abir, and Saenko, Kate. RISE: Randomized input sampling for explanation of black-box models. In *BMVC*, 2018.

Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. Why should i trust you?: Explaining the predictions of any classifier. In *KDD*, 2016.

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

Samangouei, Pouya, Kabkab, Maya, and Chellappa, Rama. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

Samek, Wojciech, Montavon, Grégoire, and Müller, Klaus-Robert. Interpreting and explaining deep models in computer vision. In *CVPR Tutorial (http://interpretable-ml.org/cvpr2018tutorial/)*, 2018.

Selvaraju, Ramprasaath R., Cogswell, Michael, Das, Abhishek, Vedantam, Ramakrishna, Parikh, Devi, and Batra, Dhruv. Grad-CAM: Why did you say that? Visual explanations from deep networks via gradient-based localization. 2017.

Shrikumar, Avanti, Greenside, Peyton, and Kundaje, Anshul. Learning important features through propagating activation differences. In *ICML*, 2017.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Springenberg, Jost Tobias, Dosovitskiy, Alexey, Brox, Thomas, and Riedmiller, Martin. Striving for simplicity: The all convolutional net. In *ICLR Workshop*, 2015.

Sundararajan, Mukund, Taly, Ankur, and Yan, Qiqi. Axiomatic attribution for deep networks. In *ICML*, 2017.

Xie, Cihang, Wang, Jianyu, Zhang, Zhishuai, Ren, Zhou, and Yuille, Alan. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2018.

Yu, Bin. Stability. *Bernoulli*, 19(4):1484–1500, 2013.

Zeiler, M.D. and Fergus, R. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

$\mathcal{L}_F$	Location		Uniform		Active		Pre-trained model
$\mathcal{I}$	Grad-CAM	LRP16	Grad-CAM	LRP16	Grad-CAM	LRP16	
Top-1 accuracy	0.71516	0.71794	0.7155	0.71656	0.70732	0.71482	0.72376
Top-5 accuracy	0.90434	0.90544	0.90526	0.90546	0.90138	0.90414	0.90876

Table 1. (Appendix) Top-1 and Top-5 accuracy for each model. The last column is the result of pre-trained model that the parameter is offered by *torchvision*. The $\mathcal{L}_F$ and $\mathcal{I}$ denotes which penalty term used for fine-tuning. Excluding the active fooling with grad-CAM interpreter, Top1 and Top5 accuracy does not harm by no more than 1% and 0.5%, respectively.