# Robust Explainability: A Tutorial on Gradient-Based Attribution Methods for Deep Neural Networks

Ian E. Nielsen[1] Ghulam Rasool[1] Dimah Dera[1,2]
Nidhal Bouaynaya[1] Ravi P. Ramachandran[1]
[1] Rowan University [2] The University of Texas Rio Grande Valley
{nielseni6, rasool, derad6, bouaynaya, ravi}@rowan.edu

**Abstract**

With the rise of deep neural networks, the challenge of explaining the predictions of these networks has become increasingly recognized. While many methods for explaining the decisions of deep neural networks exist, there is currently no consensus on how to evaluate them. On the other hand, robustness is a popular topic for deep learning research; however, it is hardly talked about in explainability until very recently. In this tutorial paper, we start by presenting gradient-based interpretability methods. These techniques use gradient signals to assign the burden of the decision on the input features. Later, we discuss how gradient-based methods can be evaluated for their robustness and the role that adversarial robustness plays in having meaningful explanations. We also discuss the limitations of gradient-based methods. Finally, we present the best practices and attributes that should be examined before choosing an explainability method. We conclude with the future directions for research in the area at the convergence of robustness and explainability.

## 1 Introduction

Deep learning (DL) has transformed the field of machine learning (ML) with deep neural networks (DNNs) being deployed in various real-world applications, including medical diagnosis, financial services, biometrics, intelligent transportation, social media, and smart home devices. Despite tremendous progress, their acceptance in mission-critical application areas is being hampered by two significant limitations. First, there is an inherent inability to explain decisions in a manner understandable to humans [1]. Second, there is vulnerability to adversarial attacks, i.e., malicious and imperceptible alterations to the input, that can fool trained networks to alter their decisions drastically [2]. These seemingly two disparate

Paper under review in IEEE Signal Processing Magazine

concepts are intrinsically linked to each other and may have their origins in the data-driven nature of DNNs with a highly nonlinear input-output relationship and over-parameterized design.

Explainability tackles the critical problem that human users cannot directly understand the complex behavior of DNNs or explain their underlying decision-making process. The explainability of ML models is the fundamental requirement for building trust with users and holds the key to their safe, fair, and successful deployment in real-world applications. The issue of explainability transcends the realm of scientific interest. The adoption of the General Data Protection Regulation (GDPR) by the European Union in May 2018 gives any citizen the "right to explanation" of an algorithmic decision made about them [3]. Explainability is both a legal right and a responsibility that has extensive social implications. The GDPR states that individuals "have the right not to be subject to a decision based solely on automated processing".

Explainability in ML is not a new topic and has been handled in many different ways, including building interpretable models or generating post hoc explanations [1]. This tutorial will focus on the latter. Given a trained neural network, either the input features are perturbed, and their effect on the network output is monitored, or a signal from the network output is back-propagated to the input. Either way, the resulting information from the perturbations or the gradient propagation provides an estimate of the contribution of input features to the output and can be presented as heatmaps.

"How good is an explanation?" is a fundamental question in explainability research. Generally, a visual analysis of the explanation is performed given the fact that these explanations are generated for humans to see and understand the behavior of the model. Recently, it has been shown that a visual analysis may not be a reliable method to ascertain the "plausibility" of an explanation [4]. However, the lack of ground-truth explanation makes it challenging to quantitatively assess, compare, and contrast various explanations.

A crucial property that all explainability methods should satisfy is insensitivity to minor input perturbations. That is, a small perturbation (possibly malicious) in the input, which does not affect network decision, should not significantly change the attributions [5, 6]. This notion of robustness is closely linked to reproducibility and replicability of

explanations. Concurrently, the explanation of a decision should change significantly when the network is under an adversarial attack, that is, imperceptible malicious changes in the input that force the network to alter its decision [2]. In an ideal world, the attribution maps should be sensitive enough to detect adversarial attack and concurrently invariant to small perturbations in the input.

This tutorial paper provides a thorough overview of gradient-based post hoc explainability methods, their attributional robustness, and the link between explainability maps and adversarial robustness. We restrict our focus on computer vision classification models, i.e., the networks are generally convolutional neural networks (CNNs) whose input data consist of images (e.g., from ImageNet datasets) and whose outputs are class scores or softmax probabilities. This tutorial is not meant to provide an exhaustive survey of all the explainability methods proposed for DL models.

## 2 Taxonomy and Definitions

Interpretability can be defined as the ability to attach a physical meaning to the prediction of a model. Along with reproducibility and replicability, interpretability falls under the larger umbrella of explainability for ML models. However, the terms interpretability and explainability are interchangeably used in the literature.

**Inherent Interpretability vs. Post hoc Explanations**: Some ML models are built to be inherently interpretable in the first place, e.g., linear models or decision trees [1]. Other models, referred to as *black box*, may require additional mathematical frameworks to explain their behavior to an audience targeting various use cases, e.g., understanding the model, debugging, providing explanations for legal purposes, or helping in decision making for downstream tasks. These mathematical frameworks, designed to explain black box models in post hoc settings, have their limitations and challenges over and above those in building ML models.

**Global vs. Local Explanations**: Based on the scope and the purpose of explanation, a user can employ a local or a global interpretability method. Global interpretability methods attempt to explain the overall decision-making process of the model, i.e., how the inputs are transformed into the output decisions at the model level. These may be more useful to re-

searchers and engineers trying to understand their models. In contrast, local interpretability methods attempt to explain specific decisions, i.e., what features of the input (e.g., pixels of an image) may have contributed (positively or negatively) to the model's output. Post hoc explanations are generally of interest to a broader audience, including but not limited to those without access to the model structure, e.g., those accessing a model-as-a-service.

The local interpretability problem can be formulated as estimating a number for each input feature that captures the effect of change in the feature value on the network output. The estimated numbers are presented as heatmaps and have the same dimension as the input features. In the literature, the terms, *attribution*, *relevance*, *importance*, *contribution*, *sensitivity*, and *saliency* scores are synonymously used.

**Feature Perturbation vs. Gradients**: Various methods for post hoc local interpretability have been proposed. Two broad categories exist, namely, methods based on feature perturbation and others based on gradient information [7]. The former class of methods perturb input features (or a set of features) by masking or altering their values, and record the effect of these changes on the network performance. In the latter case, the gradients of the output (logits or soft-max probabilities) with respect to the extracted features or the input are calculated via backpropagation and are used to estimate attribution scores. Generally, the gradients are noisy, leading to attribution maps that may show contributions from irrelevant features. Various alterations to the gradient-based approach have been proposed to handle the challenge of noise in attribution maps.

### 2.1 Requirements from Attribution Maps

Before we go into a detailed discussion about how to create local gradient-based post hoc attribution maps, it is relevant to consider what we expect from these attribution maps. Some of these requirements are defined axiomatically [8].

**Implementation Invariance**: As we know, ML models can be expressed and implemented in many different ways, mathematically or programmatically; however, two *functionally equivalent* models should produce similar output for the same input. The attribution methods must be *implementation invariant*, i.e., produce the same attribution scores for the same inputs on functionally equivalent networks, regardless of how these networks are implemented [8].

**Input Invariance**: Given the fact that neural networks are invariant to certain input transformations (e.g., a constant shift in the input), an attribution method must also be insensitive to such input transformations [9].

**Fidelity**: The fidelity or selectivity of an attribution method is linked with its ability to identify feature relevance. An attribution method with high fidelity assigns high attribution score to features that, when removed, greatly reduce network performance and vice versa [10].

**Saturation**: In the forward pass, an input feature may saturate the network, owing to the nonlinear activation function being used, e.g., the rectified linear unit (ReLU) function [8]. Consider a neural network with one ReLU, $f(x) = 1 - \text{ReLU}(1 - x)$. For all input values $x > 1$, we have: $f(x) = 1$ and $\frac{d}{dx} f(x) = 0$. Despite the fact that the input feature may change significantly, the function output stays the same and the gradient remains at zero. An attribution method must tackle the saturation in the network while estimating attributions. One possible method is to use a *reference* input or *baseline*, which can be zero (black pixel), a random number, or an average value calculated over the input dataset.

**Sensitivity**: Considering a neutral baseline (e.g., zero or black image) for all features, *sensitivity* requires that the output of the model for an input should be decomposable as the sum of the individual contributions from the input features [7]. This property is also referred to as *completeness* or *summation to delta* [8, 11]. For an attribution method to be considered sensitive, it must assign a non-zero attribution score to the single distinctive feature between two similar inputs. Furthermore, sensitivity requires that any feature, which does not affect the output of the network, must be given a zero attribution score [8].

# 3 Gradient-Based Attribution Methods

We consider a network with an $N$-dimensional input $\mathbf{x} = \{x_i\}_{i=1}^{N} \in \mathbb{R}^N$ and a $C$-dimensional output $\mathbf{S}(\mathbf{x}) = \{S_c\}_{c=1}^{C} \in \mathbb{R}^C$, where $C$ is the total number of classes and $S_c(\mathbf{x})$ represents the network's score function. Note that $S_c$ can be either a class score (logit) or soft-max probability. We use the term "gradient" for $\frac{\partial}{\partial \mathbf{x}} S_c(\mathbf{x})$. The goal of attribution methods is to estimate the attribution map, $\mathbf{A}^c = \{A_i^c\}_{i=1}^{N} \in \mathbb{R}^N$. The attribution map $\mathbf{A}^c$ captures the importance of each input feature for a specific output class $c$. In computer

vision applications, we consider CNNs with image inputs, i.e., the pixels of the image are considered input features. The resulting attribution map $\mathbf{A}^c$ has the same size $N$ as the input.

### 3.1 Gradients

We consider a linear model with $N+1$ parameters $\boldsymbol{\theta}$ and an $N$ feature input,

$$y = \theta_0 + \theta_1 x_1 + \ldots + \theta_N x_N + \epsilon = \boldsymbol{\theta}^T \mathbf{x} + \epsilon, \tag{1}$$

where $\epsilon$ is the modeling error and $\theta_0$ is the bias.

The partial derivative of the output $y$ with respect to the input $\mathbf{x}$ results in model parameters $\boldsymbol{\theta}$, which represent contributions of input features. Thus, for the linear case, model parameters serve as feature attributions.

**Saliency Maps**: Simonyan *et al.* used a similar formulation with an absolute value for constructing *Saliency maps* for DNNs, $\mathbf{A}^c_{\text{Saliency}} = \left|\frac{\partial S_c}{\partial \mathbf{x}}\right|_{\mathbf{x}_0}$, where $\mathbf{x}_0$ is the input [12]. It is important to highlight that $S_c$ is a nonlinear function of the input $\mathbf{x}$ and thus, in contrast to the linear case, the model parameters no more represent feature attributions. It has been shown that saliency maps represent the first-order approximation of the attributions [12]. The major challenge with saliency maps is that they are visually noisy and a great deal of research has focused on removing noise and improving visualization [13].

**Deconvolutional Networks (DeconvNets)**: Saliency maps are closely related to DeconvNets proposed by Zeiler and Fergus [14]. In saliency maps, the gradient signals are zeroed during backpropagation at each ReLU when the input to the same ReLU was negative during forward pass. In contrast, DeconvNet reduces the negative gradients to zero at each ReLU, ignoring the fact whether the input to the same ReLU was negative or positive during the forward propagation.

**Guided Backpropagation (GBP)**: GBP combines operations from both saliency maps and DeconvNet [15]. That is, during backpropagation, the attribution signal is reduced to zero at a ReLU when either the gradient signal itself is negative or the input to the ReLU at the time of the forward pass was negative. Removing negatively contributing features may reduce noise and improve visualization of attribution maps in some cases.

**SmoothGrad**: SmoothGrad reduces noise and visual diffusion by averaging over explanations generated for multiple noisy copies of the input $\mathbf{x}$ [13]. For a saliency map $\mathbf{A}^c$ calculated for the input $\mathbf{x}$, SmoothGrad is given by $\mathbf{A}^c_{\text{sg}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{A}^c\left(\mathbf{x} + \mathcal{N}(0, \sigma^2)\right)$, where $n$ is the number of samples and $\mathcal{N}$ represents the Gaussian distribution.

**Gradient⊙Input**: In Gradient⊙Input, the attribution scores are calculated by element-wise multiplication of gradients with the input, i.e., $\mathbf{A}^c_{\text{Gradient}\odot\text{Input}} = \frac{\partial S_c(\mathbf{x})}{\partial \mathbf{x}} \odot \mathbf{x}$.

The element-wise multiplication can be considered as an application of a model-independent filter (the input), which may reduce noise and smoothen the attribution maps [7].

**Integrated Gradients (IG)**: IG can be considered a smoother version of Gradient⊙Input, specifically designed to satisfy two axioms of explainability, i.e., sensitivity and implementation invariance [8]. IG along the $i^{th}$ dimension for an input $x$ and baseline $\acute{x}$ is given by $\text{IG}_i(x) = (x - \acute{x}) \int_0^1 \frac{\partial}{\partial x_i} S_c\left(\acute{x} + \alpha(x - \acute{x})\right) d\alpha$.

IG calculates the average of all gradients along a straight line between the baseline and the input. In practice, we can only use a finite number of samples to approximate the integral, which may introduce an approximation error.

### 3.2 Attribution Propagation

Attribution propagation can be considered as an alternative to calculating gradients. Recursively, attribution propagation methods decompose the decision made by the network into contributions from previous layers, all the way to the input. These methods use forward-pass activations (starting with the activation of the neuron in the last layer) to move back layer-by-layer in the network and distribute the burden of the decision over the input features. This class of methods includes various forms of Layer-wise Relevance Propagation (LRP), Deep Taylor Decomposition [16], and Deep Learning Important FeaTures (DeepLIFT) [11]. These methods do not strictly use *gradients* intrinsically; however, their relationship to Gradient⊙Input has been mathematically established [7].

**Layer-Wise Relevance Propagation (LRP)**: LRP propagates relevance scores from the last layer of the network to the input using the "conservation property" [17]. That is, what was received by a neuron in the forward pass (activations) must be redistributed to the lower layer (a layer nearer to the input) by an equal amount. Going from the output to the input, layer-by-layer, the relevance scores are scaled at each layer using the information

from the forward pass. LRP starts with the activation of the neuron in the last layer. Let $j$ and $k$ be neurons at two consecutive layers $l$ and $l+1$, with layer $l$ closer to the input. Let $\theta_{jk}$ be the learnable parameters that connect both layers. The neuronal activation $a_k^{[l+1]}$ in the forward pass is defined as $a_k^{[l+1]} = \text{ReLU}\left(\sum_j a_j^{[l]}\theta_{jk}\right)$. Given that we have $r_k^{[l+1]}$, i.e., relevance score at layer $l+1$, we can calculate the relevance score $r_j^{[l]}$ at layer $l$ using:

$$r_j^{[l]} = \sum_k \frac{a_j^{[l]}\theta_{jk}}{\epsilon + \sum_j a_j^{[l]}\theta_{jk}} r_k^{[l+1]}. \tag{2}$$

Equation 2 is called LRP-$\epsilon$, where $\epsilon$ is added to absorb some relevance when the contributions to the activation of neuron $k$ are weak or contradictory. When $\epsilon \gg \sum_j a_j^{[l]}\theta_{jk}$, only the most salient explanation factors survive the absorption, leading to noise reduction and sparser explanations [17]. It has been shown that for CNNs with ReLU activation functions, LRP-$\epsilon$ implements a slightly modified form of Gradient$\odot$Input, where the gradients are normalized at each layer by the activations $\sum_j a_j^{[l]}\theta_{jk}$ [7]. Montavon *et al.* proposed Deep Taylor Decomposition, which provided theoretical foundations for LRP using Taylor series approximation [16].

**Deep Learning Important FeaTures (DeepLIFT)**: DeepLIFT was designed to tackle the saturation problem using "reference activations", calculated in the forward pass with the baseline input [11]. DeepLIFT compares the activation of each neuron to its reference activation and assigns contribution scores according to the difference [11]. It has been shown that DeepLIFT (Rescale rule) is equivalent to Gradient$\odot$Input [7].

### 3.3 Gradient-weighted Class Activation Mapping (Grad-CAM)

Grad-CAM uses the class-specific gradient information flowing into the final convolutional layer of a CNN to produce a coarse localization map of the important features of the input [18]. Grad-CAM analyzes which regions are activated in the feature maps of the last convolutional layer. Grad-CAM can be combined with GBP, referred to as Guided Grad-CAM, to improve pixel-level granularity of attribution maps. In Fig. 1, we present attribution maps generated using various methods.

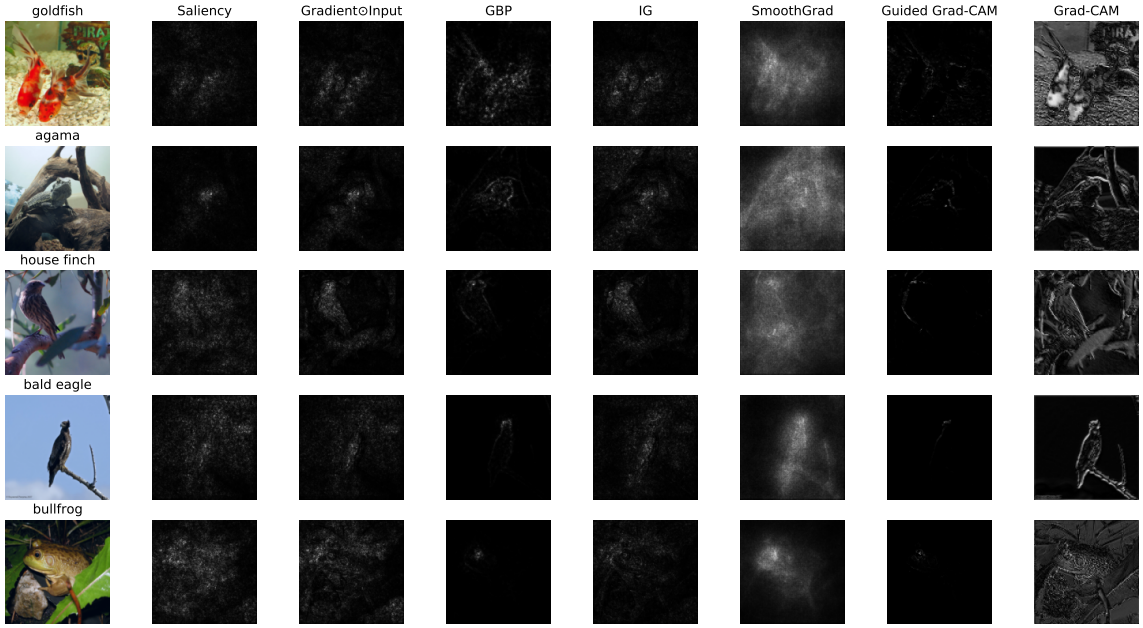


Figure 1: Attribution maps generated using different methods are presented. The first column presents test images from ImageNet and rest of the columns present attribution maps estimated using various methods. Abbreviations used: GBP - Guided Backpropagation, IG - Integrated Gradients, and Grad-CAM - Gradient-weighted Class Activation Mapping.

# 4 Analysis of Gradient-Based Attribution Methods

Attribution maps are designed to explain the decisions made by ML models. However, a large body of research has found that various approaches to create these attributions have their own limitations [9, 4, 6].

Starting with measures that can be used to evaluate explanations in DL models, we provide a detailed analysis of the performance of these methods and their limitations.

## 4.1 Evaluation of Attribution Maps

"How good is an explanation?" is one of the fundamental questions in ML explainability research. The lack of ground truth explanations makes it challenging to validate attribution methods. Ideal evaluation of these methods will depend upon fully knowing the process of how the ML model reached its decision - the very problem that we are trying to solve. Furthermore, it is hard to disentangle the errors made by models from the errors made by the attribution methods [8]. Given that the notion of explanation is centered around human visual perception, the predominant evaluations of attributions have been subjective. However, objective evaluation is equally, or perhaps more, important to establish rigorous

theoretical foundations, compare and contrast various approaches, and improve upon these methods [19].

**Visual Evaluation**: A visual analysis of the attribution maps may seem to be the most plausible way of evaluation as these are created to explain the behavior of DL models to human operators and designers. Visual analysis includes qualitative displays of explanation examples, crowd-sourced evaluations of human satisfaction with the explanations, as well as whether humans are able to understand the model output [19]. However, it may be misleading to rely solely on visual analysis for determining whether an attribution method is able to capture the features that a network considers important [4]. A visual analysis may bias the evaluation of how humans understand the phenomenon and make decisions, rather than capturing how the network reached a particular decision. This may hold true especially for the methods that multiply the input with the gradient, i.e., Gradient$\odot$Input and IG.

**Feature Perturbation-based Evaluation**: Removing the most important features identified by an attribution method and recording its effect on the performance of the network may provide an objective approach to evaluate various attribution methods [20]. A good attribution method will identify the most important pixels, which when removed should maximally degrade the network performance. The metric is referred to as the Most Relevant First (MoRF). As the network input size is fixed, the removed pixels are replaced with either the average value (calculated over the input dataset), zero (i.e., black pixel), or random values [10]. It is obvious that replacing pixels with an average value or black pixels can introduce high-frequency edges, which may degrade network performance - unrelated to the removal of important pixels [21]. We may choose to remove the least important pixels first - thus partially decoupling the effects of artifacts introduced by high-frequency edges from those caused by removing important pixels. The metric is referred to as the Least Relevant First (LeRF) [21].

A recent study by Tomsett *et al.* evaluated the reliability of both MoRF and LeRF using four different statistical tests from the psychometric literature [10]. These tests included inter-rater reliability, inter-method reliability, internal consistency reliability, and test-retest reliability, where each image corresponded to a different rater and methods in-

cluded different attribution map generation techniques. Both MoRF and LeRF showed: (1) high variance across all tested images, (2) sensitivity to whether the removed pixels were replaced with the mean of the dataset or random values, (3) low inter-rater reliability, i.e., the rankings of different methods were highly inconsistent, and (4) low correlation with each other. The results were reported for the classification task using the CIFAR-10 dataset. The absence of ground truth explainability and the limited testing (using one dataset only) makes it hard to generalize these results to other metrics. However, the study raised important questions about the validity of different metrics that are extensively used in the explainability literature.

**Remove and Retrain (ROAR)**: Replacing pixels from the input image may change the distribution of the data that were used to train the network, thus violating the assumption that training and evaluation data must have the same distribution [22]. In remove and retrain (ROAR), the model is retrained and evaluated every time after removing a set of most important pixels. ROAR is computationally expensive and does not address the question of validity of explanation for each input, rather evaluates the method globally over the whole dataset. Furthermore, the retraining strategy may force the network to learn from the features that were not present in the original dataset (e.g., high-frequency edges introduced due to pixel replacement). This leads to evaluating a new model with newly learned parameters, not the original model.

**(In)fidelity and Sensitivity**: (In)fidelity quantifies the statistically expected difference between (1) the dot product of the input perturbation to the attribution scores and (2) the output perturbation (difference in the score function $S_c(\mathbf{x})$ values after *significant perturbations* introduced in the input $\mathbf{x}$) [19]. (In)fidelity allows for a number of significant perturbations, including random and non-random perturbations that lead the input towards a predefined single or multiple baseline values. Random perturbations with a small amount of additive Gaussian noise allows the measure to be robust to small mis-specifications or noise in either the test input or the reference point [19]. On the other hand, the "sensitivity" measures the degree to which the explanation is affected by *insignificant perturbations* in the test point [19]. A good attribution method will exhibit low sensitivity, i.e., producing same explanations for minor variations in the input.

**Sanity Checks**: Recently, Adebayo *et al.* introduced two *sanity checks* for evaluating the sensitivity of attribution methods to the model parameters and the dataset [4]. The first check consists of replacing all the learned parameters of the network with random numbers. The resulting attribution maps are compared to the original maps using various correlation metrics to ascertain whether the attribution maps were able to capture the changes in the network parameters. The second check evaluates attribution methods on networks trained using randomly permuted labels. The attribution maps which remain unchanged for either of these checks are considered to have failed.

### 4.2 Limitations of Attribution Maps

The attribution methods explain the behaviour of a model for a single test point selected from the evaluation dataset. The explanation provided by the attribution methods for the selected single point may be too brittle and could lead to a false conclusion about the performance of the model [5]. Thus, understanding a complex model with a single or even multiple pointwise explanations without theoretically grounded metrics is perhaps too optimistic [5]. Explaining a model at a single point and then generalizing to the whole dataset is an open question for the research community.

**Class Agnostic Behaviour**: In some cases, the attribution maps may remain the same regardless of the class chosen by the user to compute the gradients. That is, for a given input image, similar attribution maps are generated despite the fact that the class label is changed, e.g, the network is forced to predict a certain class as in adversarial attacks [1]. In Fig. 2, we present attribution maps corresponding to 7 different methods generated for different target classes using the same input image. It is evident that most of these methods, except Grad-CAM, are not class sensitive. A similar behavior was observed when neural networks were trained using a dataset with permuted class labels. Many state-of-the-art attribution methods (except saliency maps and SmoothGrad) generated explanations that were insensitive to the permuted class labels. In summary, these methods are not able to capture the relationship between network input and output, and thus generate the same explanations, even if the class labels are changed.

**Insensitivity to Model Parameters**: Attribution maps should be sensitive to the learned optimal network parameters. That is, if the parameters of a trained network are replaced

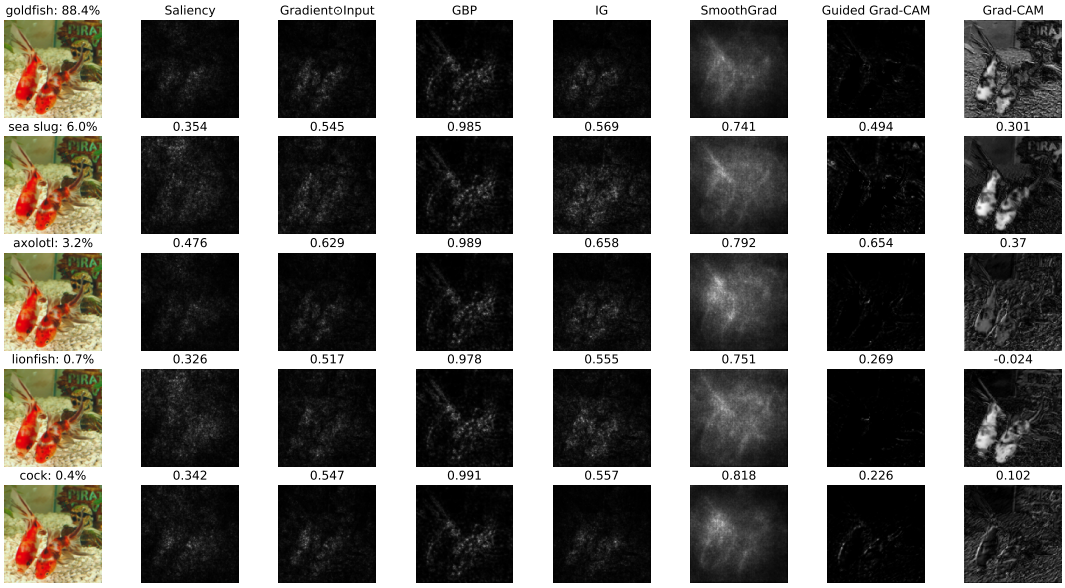


Figure 2: The class agonist behavior of attribution methods is presented. The first column present input image and other columns show attribution maps generated by different methods. The target class and soft-max probability values are shown on the top of image in the first column. The number on the top of attribution maps are Spearman rank correlation values, calculated between the attribution maps of the true class (top row) and the target class. High correlation values show that the method is not class discriminatory, i.e., the attribution maps for any choice of class label are correlated.

by random numbers, attribution maps should capture the effect of this change. However, Adebayo *et al.* found that GBP and Guided Grad-CAM were insensitive to the learned parameters in the top layers (near to the output) [4].

**Sensitivity to Input Transformations**: The explanations may be sensitive to factors that do not contribute to the model prediction, e.g., a constant shift in the input [9]. Gradient⊙Input and other methods (e.g., IG) that use input in the computation of attributions are generally sensitive to such input transformations. For the case of IG, this may further depend on the chosen input baseline [9]. Saliency maps, DeconvNet, and GBP were found to be insensitive to such transformations as these methods rely solely on the network parameters (no multiplication by the input) to generate attribution maps.

**Input Dominance**: Gradient⊙Input, DeepLIFT, and IG multiply the input with gradients to leverage the information present in the input features. This may help reduce noise in the attribution maps and produce more human interpretable explanations. However, in some cases, the attribution maps generated by these methods may be dominated by the input.

The input does not depend on the network and cannot capture how the network processed data to make a decision [4].

**Partial Input Recovery**: GBP and DeconvNet can be considered as variants of saliency maps with different rules governing negative gradients at ReLUs. These methods are able to generate relatively more human-interpretable visualizations due to the backward ReLU (used by both GBP and DeconvNet) and the local connections in CNNs. Nie *et al.* showed that both GBP and DeconvNet performed (partial) input image recovery, a phenomenon that is unrelated to the network decisions [23].

**Input Baseline**: DeepLIFT and IG use an input baseline to improve attributions. A reasonable choice of baseline depends upon the domain and task at hand. An uninformed and inappropriate choice of baseline may invalidate the explainability provided by the attribution method [9].

**Sensitivity to Hyperparameters**: The explanations generated by some methods, e.g., SmoothGrad or IG may depend on the chosen hyperparameters, e.g., the number of samples used [24].

# 5 Explainability and Robustness

Until recently, the explainability of DL models and their robustness were being studied in isolation [25]. However, recent work has provided a strong link between these two apparently disparate aspects of DL models. Before we explore these ideas any further, it is important to highlight two types of robustness in explainability, i.e., (1) the robustness of attribution maps, and (2) the robustness of DL models to adversarial attacks, which is intrinsically linked to their explainability.

## 5.1 Attributional Robustness

Attributional robustness is related to the stability of an attribution map in the face of a small perturbation in the input caused by natural reasons (e.g., data distribution shift) or introduced by an adversary [6, 26, 27]. It was shown that the input can be adversarially manipulated to change the attribution maps without affecting the network performance, i.e., the prediction of the network does not change [6, 26, 27]. Recent research attributes the origin of these false and manipulated explanations to the vulnerabilities of the neural

network, e.g, non-smooth decision boundaries, and not the attribution generation methods [26, 27].

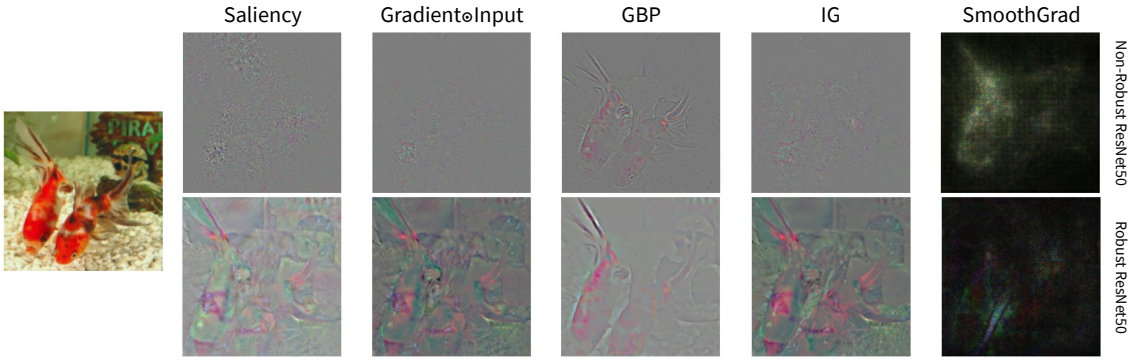


Figure 3: Input image and saliency maps generated for two different network are presented. (Left) Input image. (Top row) ResNet50 trained on natural dataset. (Bottom row) ResNet50 robustly trained using Projected Gradient Descent (PGD) attacks [25]. It is evident that attribution maps generated for adversarially trained network are more visually appealing.

### 5.2 Adversarial Robustness

Neural networks are known to be vulnerable to "smart noise" or adversarial attacks. These attacks are quasi-imperceptible perturbations in the input, measured using $L_p$ norms, that force a network to change its output [2]. Currently, adversarial training is the most common strategy that may provide limited defence against known attacks. In adversarial training, a modified objective function is optimized which helps in adversarial robustness by increasing the level of perturbation required to successfully change the network decision [2]. With $\boldsymbol{\theta}$ denoting the learnable parameters of the model, training data $(x, y) \sim D$, and perturbation $\delta \in \Delta$, adversarial training can be formulated as the following min-max optimization problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{x,y\sim D} \left[ \max_{\delta \in \Delta} \mathcal{L}(x + \delta, y; \boldsymbol{\theta}) \right], \tag{3}$$

where $\mathcal{L}$ denotes the model's loss function.

The adversarial training can be considered as a method for the model to learn certain ($l_p$-bounded) invariances to the dataset. Some recent studies have established that learning

certain types of invariances qualitatively (visually) and quantitatively may improve attribution maps, i.e., maps look more relevant to the object as viewed by a human operator [25, 28]. In a way, adversarial training helps the network learn more like the human visual system learns. Figure 3 shows the difference between attribution maps generated for adversarially and naturally trained network. It is evident that the attribution maps produced by adversarially trained models seem more visually aligned with human perception [29, 25, 30, 28]. Tsipras *et al.* described this relationship between adversarial robustness and enhanced visual alignment of attribution maps as an "unexpected benefit" of adversarial training [25]. Later, a number of studies verified and made an effort to explain the natural connection between adversarial robustness and explainability [28, 30, 31].

Etmann *et al.* showed that the improved interpretability of the saliency maps of a robustified neural network was not a side-effect of adversarial training, but a general property enjoyed by networks that are robust to adversarial perturbations [28]. The authors showed that robustness could be defined as the distance of a test point to its closest decision boundary, and that increasing the distance (robustness) resulted in an increased alignment between the input and its attribution map.

Recently, Kim *et al.* showed that the gradients from adversarially trained networks were better aligned with the human visual system as the adversarial training caused the gradients to lie closer to the image manifold [30]. They also reported differences in the attribution maps generated with robust networks trained using $l_2$ and $l_\infty$ adversarial images. The neural networks trained with $l_2$ were more effective at emphasizing important features while attributions from $l_\infty$-trained networks were better at identifying less important features.

Ignatiev *et al.* performed a theoretical analysis using a generalized form of hitting set duality to relate explanations and adversarial examples [31]. The authors proposed the dual concept of counterexamples (and adversarial examples) and the notion of breaking an explanation. They established that each explanation must break every counterexample and vice versa. Thus, concluding that the more counterexamples (adversarial examples) the model explains, the better the interpretability of the model.

## 6 Best Practices for the Community

Explainability of black box machine learning models is important for multiple reasons, including understanding the internal workings of these models. Attribution methods are in themselves a set of mathematical operations with certain assumptions and may add another layer of abstraction over the goal of understating data and making predictions. While analysing explanations, it is also not clear how to disentangle errors in the explanation method from errors in the DL model. Currently, there is no consensus on which methods are better than others at explaining network predictions. However, there are some considerations that should be made when choosing attribution methods.

**Gradient-based vs. Perturbation Methods**: Gradient-based methods are computationally less expensive as in some cases these may require only one forward and one backpropagation step for estimating attributions. Perturbation-based methods generally solve an optimization problem and thus may require multiple forward passes through the network. Furthermore, gradient-based methods are more robust to input perturbations as compared to perturbation-based methods and should be preferred when robustness is a priority for the user [5].

**Efficiency**: The efficiency of an attribution method can be related to the number of passes (forward and backward) through the network. Saliency maps, Input$\odot$Gradient, GBP, and Grad-CAM require one forward and one backpropagation step. IG and SmoothGrad may require 50 to 200 steps depending upon the problem domain, dataset, and the scope of explanation.

**Input Baseline**: Some attribution methods require an input baseline, which acts as the absence of the feature from the input. The baseline can be zero (a black image), an average value calculated from the dataset, a blurred version of the input image, or random values generated with Gaussian, uniform or other distributions. The choice of baseline can significantly alter the explanation [32, 9]. Since there is no current consensus on which baseline is optimal, it is difficult to recommend the use of these methods as an accurate way to explain model predictions.

**Human Interpretability**: Relying solely on visual analysis for understanding and comparing attribution maps can be unreliable [4]. Some attribution maps may seem visually appealing, but they may not actually help us interpret model predictions. On the other hand, there is still no consensus in the community on the reliability of the various metrics to use for comparing attribution maps. Finally, a typical attribution method tends to consider each pixel as the fundamental unit explanation, which is not the basic unit used in human perception. Some recent studies have pointed out a limited usefulness of the current model explanation methods (e.g., heatmaps) and highlighted the need for a deeper investigation into the methods for presenting interpretations of models to human operators [33].

# 7 Conclusion and Future Research Directions

We have presented an overview of the post hoc gradient-based attribution methods to explain the decisions made by deep neural networks. These techniques represent a small but very significant part of a large body of methods that focus on explaining black box ML models. These methods are fast and can provide explanations that are robust as compared to other approaches. In some cases, these explanations may seem convincing. However, they should be approached with caution due to the inherent limitations of these methods as discussed above. Application of these methods in real-world settings, without comprehending their limitations, can create a false sense of confidence in ML decision-making.

We consider that the robustness of interpretability methods is tightly coupled with the robustness of the models being explained. This area needs research efforts on both fronts, empirical as well as theoretical. There is a need to bring research communities from explainability and robustness together to explore these questions. Finally, given the state-of-the-art in post hoc explainability methods and their vulnerabilities and limitations, there is a need in the ML community to focus on building models that are inherently explainable, but as versatile, efficient, accurate and scalable as deep neural networks.

# 8 Acknowledgements

This research was supported by National Science Foundation Awards ECCS-1903466 and OAC-2008690, and US Dept of Education GAANN award P200A180055

## References


[1] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *International Conference on Learning Representations*, 2018.

[3] B. Goodman and S. Flaxman, "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"," *AI magazine*, vol. 38, no. 3, pp. 50–57, 2017.

[4] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity Checks for Saliency Maps," *Advances in Neural Information Processing Systems*, vol. 31, pp. 9505–9515, 2018.

[5] D. Alvarez-Melis and T. S. Jaakkola, "On the Robustness of Interpretability Methods," *arXiv preprint arXiv:1806.08049*, 2018.

[6] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of Neural Networks is Fragile," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3681–3688, 2019.

[7] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Gradient-based Attribution Methods," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 169–191, Springer, 2019.

[8] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," in *International Conference on Machine Learning*, pp. 3319–3328, PMLR, 2017.

[9] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, "The (Un)reliability of Saliency Methods," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 267–280, Springer, 2019.

[10] R. Tomsett, D. Harborne, S. Chakraborty, P. Gurram, and A. Preece, "Sanity Checks for Saliency Metrics," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 6021–6029, 2020.

[11] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning Important Features Through Propagating Activation Differences," in *International Conference on Machine Learning*, pp. 3145–3153, PMLR, 2017.

[12] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *arXiv preprint arXiv:1312.6034*, 2013.

[13] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: Removing Noise by Adding Noise," *arXiv preprint arXiv:1706.03825*, 2017.

[14] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *European Conference on Computer Vision*, pp. 818–833, Springer, 2014.

[15] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," *arXiv preprint arXiv:1412.6806*, 2014.

[16] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining Nonlinear Classification Decisions with Deep Taylor Decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[17] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," in *IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

[19] C.-K. Yeh, C.-Y. Hsieh, A. S. Suggala, D. I. Inouye, and P. Ravikumar, "On the (In)fidelity and Sensitivity for Explanations," *arXiv preprint arXiv:1901.09392*, 2019.

[20] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the Visualization of What a Deep Neural Network Has Learned," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2660–2673, 2016.

[21] S. Srinivas and F. Fleuret, "Full-Gradient Representation for Neural Network Visualization," *arXiv preprint arXiv:1905.00780*, 2019.

[22] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A Benchmark for Interpretability Methods in Deep Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 9737–9748, 2019.

[23] W. Nie, Y. Zhang, and A. Patel, "A Theoretical Explanation for Perplexing Behaviors of Backpropagation-based Visualizations," in *International Conference on Machine Learning*, pp. 3809–3818, PMLR, 2018.

[24] N. Bansal, C. Agarwal, and A. Nguyen, "SAM: The Sensitivity of Attribution Methods to Hyperparameters," in *CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 11–21, 2020.

[25] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness May Be at Odds with Accuracy," in *International Conference on Learning Representations*, 2019.

[26] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations Can be Manipulated and Geometry is to Blame," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 13589–13600, 2019.

[27] D. Lim, H. Lee, and S. Kim, "Building Reliable Explanations of Unreliable Neural Networks: Locally Smoothing Perspective of Model Interpretation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6468–6477, 2021.

[28] C. Etmann, S. Lunz, P. Maass, and C. Schoenlieb, "On the Connection Between Adversarial Robustness and Saliency Map Interpretability," in *International Conference on Machine Learning*, pp. 1823–1832, PMLR, 2019.

[29] A. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[30] B. Kim, J. Seo, and T. Jeon, "Bridging Adversarial Robustness and Gradient Interpretability," *arXiv preprint arXiv:1903.11626*, 2019.

[31] A. Ignatiev, N. Narodytska, and J. Marques-Silva, "On Relating Explanations and Adversarial Examples," *Advances in Neural Information Processing Systems*, vol. 32, pp. 15883–15893, 2019.

[32] P. Sturmfels, S. Lundberg, and S.-I. Lee, "Visualizing the Impact of Feature Attribution Baselines," *Distill*, vol. 5, no. 1, p. e22, 2020.

[33] E. Chu, D. Roy, and J. Andreas, "Are Visual Explanations Useful? A Case Study in Model-in-the-loop Prediction," *arXiv preprint arXiv:2007.12248*, 2020.