

---

# The (Un)reliability of saliency methods

---

Pieter-Jan Kindermans<sup>†</sup>, Sara Hooker<sup>†</sup>, Julius Adebayo

Google Brain\*

{pikinder, shooker}@google.com

Maximilian Alber, Kristof T. Schütt, Sven Dähne

TU-Berlin

Dumitru Erhan, Been Kim

Google Brain

## Abstract

Saliency methods aim to explain the predictions of deep neural networks. These methods lack reliability when the explanation is sensitive to factors that do not contribute to the model prediction. We use a simple and common pre-processing step—adding a constant shift to the input data—to show that a transformation with no effect on the model can cause numerous methods to incorrectly attribute. In order to guarantee reliability, we posit that methods should fulfill input invariance, the requirement that a saliency method mirror the sensitivity of the model with respect to transformations of the input. We show, through several examples, that saliency methods that do not satisfy input invariance result in misleading attribution.

## 1 Introduction

While considerable research has focused on discerning the decision process of neural networks [2, 11, 3, 16, 13, 1, 15, 7, 6, 17, 14, 12, 5], there remains a trade-off between model complexity and interpretability. Research to address this tension is urgently needed; reliable explanations build trust with users, help identify points of model failure and remove barriers to entry for the deployment of deep neural networks in domains like health care, security and transportation.

In deep neural networks, data representation is delegated to the model and subsequently we cannot generally say in an informative way what led to a model prediction. Instead, saliency methods aim to infer insights about the  $f(x)$  learnt by the model by ranking the explanatory power of constituent inputs. While unified in purpose, these methods are surprisingly divergent and non-overlapping in outcome. Evaluating the reliability of these methods is complicated by a lack of ground truth, as ground truth would depend upon full transparency into how a model arrives at a decision—the very problem we are trying to solve for in the first place.

Given the need for a quantitative method of comparison, several properties such as completeness, implementation invariance and sensitivity have been articulated as desirable to ensure that saliency methods are reliable [1, 14]. Implementation invariance, proposed as an axiom for attribution methods by [14], is the requirement that functionally equivalent networks (models with different architectures but equal outputs for all inputs), always attribute in an identical way.

This work posits that a second invariance axiom, which we term *input invariance*, needs to be satisfied to ensure reliable interpretation of the input’s contribution to the model prediction. *Input invariance* requires that the saliency method mirror the sensitivity of the model with respect to

---

\*Work done as part of the Google Brain Residency program, Pieter-Jan Kindermans and Sara Hooker have contributed equally.

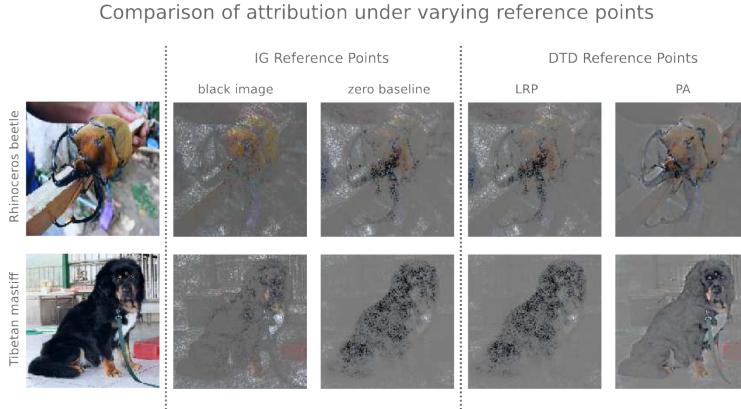


Figure 1: Integrated gradients and Deep Taylor Decomposition determine input attribution relative to a chosen reference point. This choice determines the vantage point for all subsequent attribution. Using two example reference points for each method we demonstrate that changing the reference causes the attribution to diverge. The attributions are visualized in a consistent manner with the IG paper [14]. Visualisations were made using ImageNet data. [8] and the VGG16 architecture [11].

transformations of the input. We demonstrate that numerous methods do not satisfy input invariance using a simple transformation – a constant shift of the input – that changes the attribution of these methods but does not affect the model prediction or weights. Our results demonstrate that explanations of a networks predictions can be purposefully manipulated using surprisingly simple transformations to be misleading. This work is motivated by an understanding that saliency methods are highly valued tools for gaining intuition about a network. Determining points of failure is a necessary step for knowledgeable use of these tools as well as a pre-requisite for domains like medicine where the incorrect classification of an input as salient carries a high cost.

In this work we:

- introduce the axiom *input invariance* and show, using a simple constant shift in the input, that certain saliency methods do not satisfy this property. (See Fig. 3).
- demonstrate using MNIST that we can purposefully force misleading attribution (See Fig. 4 and Fig. 5).
- show that "reference point" methods – Integrated gradients and the Deep Taylor Decomposition– have diverging attribution satisfy input invariance contingent on the choice of reference and the type of transformation considered (See Fig. 1).
- propose data normalization as a way to ensure that some methods satisfy input invariance for the type of transformation considered. Discuss the need for wider research as normalization does not systematically guarantee reliable attribution for all possible transformations.

In **Section 2**, we detail our experiment framework. In **Section 3**, we determine that while the model is invariant to the input transformation considered, several saliency methods attribute to the mean shift. In **Section 4** we discuss "reference point" methods and illustrate the importance of choosing an appropriate reference before discussing some directions for future research in **Section 5**.

## 2 The Model is Invariant to a Constant Shift in Input

We show that, by construction, the bias of a neural network compensates for the constant shift resulting in two networks with identical weights and predictions.

We compare the attribution across two networks,  $f_1(x)$  and  $f_2(x)$ .  $f_1(x)$  is a network trained on input  $x_1^i$  that denotes sample  $i$  from training set  $X_1$ . The classification task of network 1 is:

$$f_1(x_1^i) = y^i,$$

$f_2(x)$  is a network that predicts the classification of a transformed input  $\mathbf{x}_2^i$ . The relationship between  $\mathbf{x}_1^i$  and  $\mathbf{x}_2^i$  is the addition of constant vector  $\mathbf{m}_2$ :

$$\forall i, \mathbf{x}_2^i = \mathbf{x}_1^i + \mathbf{m}_2.$$

Network 1 and 2 differ only by construction. Consider the first layer neuron before non-linearity in  $f_1(x)$ :

$$z = \mathbf{w}^T \mathbf{x}_1 + b_1.$$

We alter the biases in the first layer neuron by adding the mean shift  $\mathbf{m}_2$ . This now becomes Network 2:

$$b_2 = b_1 - \mathbf{w}^T \mathbf{m}_2.$$

As a result the first layer activations are the same for  $f_1(x)$  and  $f_2(x)$ :

$$z = \mathbf{w}^T \mathbf{x}_2 + b_2 = \mathbf{w}^T \mathbf{x}_1 + \mathbf{w}^T \mathbf{m}_2 + b_1 - \mathbf{w}^T \mathbf{m}_2.$$

Note that the gradient with respect to the input remains unchanged as well:

$$\frac{\partial f_1(\mathbf{x}_1^i)}{\partial \mathbf{x}_1^i} = \frac{\partial f_2(\mathbf{x}_2^i)}{\partial \mathbf{x}_2^i}.$$

We have shown that Network 2 cancels out the mean shift transformation. This means that  $f_1(x)$  and  $f_2(x)$  have identical weights and produce the same output for all corresponding samples,  $\mathbf{x}_1^i \in X_1$ ,  $\mathbf{x}_2^i \in X_2$ :

$$\forall i, f_1(\mathbf{x}_1^i) = f_2(\mathbf{x}_2^i).$$

## 2.1 Experimental Setup

Now, we describe our experiment setup to evaluate the input invariance of a set of saliency methods. Most saliency research to date has centered on convolutional neural networks (CNN). In this work, we also evaluate input invariance using a CNN. Network 1 is a 3 layer multi-layer perceptron with 1024 ReLu-activated neurons each. Network 1 classifies MNIST image inputs in a  $[0,1]$  encoding. We consider a negative constant shift of  $\mathbf{m}_2 = -1$ ; Network 2 classifies MNIST image inputs in a  $[-1,0]$  MNIST encoding. The first network is trained for 10 epochs using mini-batch stochastic gradient descent (SGD). The final accuracy is 98.3% for both<sup>3</sup>. In 3.1 we introduce the saliency methods we evaluate.

## 3 The (in)sensitivity of saliency methods to mean shifts

In 3.1 we introduce key approaches to the classification of inputs as salient and the saliency methods we evaluate. In 3.2 we find that gradient and signal methods satisfy input invariance. In 3.3 we find that all attribution methods considered have points of failure.

### 3.1 Saliency Methods Considered

Saliency methods broadly fall into three different categories:

1. **Gradients (Sensitivity)** [2, 10]) shows how a small change to the input affects the classification score for the output of interest.
2. **Signal methods** such as DeConvNet [16], Guided BackProp [13] and PatternNet [5] aim to isolate input patterns that stimulate neuron activation in higher layers.

---

<sup>3</sup>Although there is a gap between this and the state of art, the gap does not significantly influence our findings.

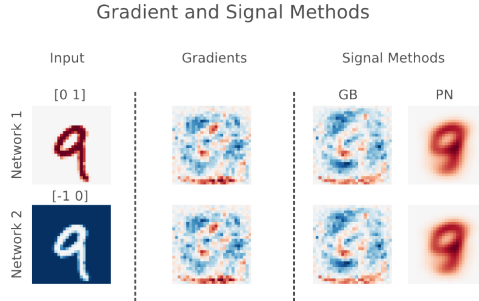


Figure 2: Evaluating the sensitivity of gradient and signal methods using MNIST with a [0,1] encoding for network  $f_1$  and a [-1,0] encoding for network  $f_2$ . Both raw gradients and signal methods satisfy input invariance by producing identical saliency heatmaps for both networks.

- Attribution methods** such as Deep-Taylor Decomposition [6] and Integrated Gradients [14] assign importance to input dimensions by decomposing the value  $y_j$  at an output neuron  $j$  into contributions from the individual input dimensions:

$$s_j = A(\mathbf{x})_j.$$

$s_j$  is the decomposition into input contributions and has the same number of dimensions as  $\mathbf{x}$ ,  $A(\mathbf{x})_j$  signifies the attribution method applied to output  $j$  for sample  $\mathbf{x}$ . Attribution methods are distinct from gradients because of the insistence on *completeness*; the sum of all attributions should be approximately equal to the original output  $y_i$ .

We consider the input invariance of each category separately (by evaluating raw gradients, Guided-Backprop, PatternNet, Integrated Gradients and Deep Taylor Decomposition).

The experiment setup and methodology is as described in **Section 2**. Each method is evaluated by comparing the saliency heatmaps for the predictions of network 1 and 2, where  $\mathbf{x}_2^i$  is simply the mean shifted input ( $\mathbf{x}_1^i + \mathbf{m}_2$ ). A saliency method that satisfies input invariance will produce identical saliency heatmaps for Network 1 and 2 despite the constant shift in input.

### 3.2 Gradient and Signal Methods Satisfy Input Invariance

Gradient and signal methods are not sensitive to a constant shift in inputs. In Fig. 2 raw gradients, PatternNet (PN), [5] and GuidedBackprop (GB) [13] produce identical saliency heatmaps for both networks. Intuitively, gradient, PN and GB satisfy input invariance given that we are comparing two networks with an identical  $f(x)$ . All three methods determine attribution entirely as a function of the network/pattern weights and thus will be input invariant as long as we are comparing networks with identical weights.

In the same manner, we can say that these methods will not be input invariant when comparing networks with different weights (even if we consider models with different architectures but identical predictions for every input).

### 3.3 The Sensitivity of Attribution Methods

We evaluate the following attribution methods: gradient times input (GI), integrated gradients (IG, [14]) and the deep-taylor decomposition (DTD, [6]).

In 3.3.1 we find GI to be sensitive to constant shifts in the input. In 3.3.2 we group discussion of IG and DTD under "reference point" methods because both require that attribution is done in relation to a chosen reference. We find that satisfying input invariance depends upon the choice of reference point and the type of constant shift to the input.

#### 3.3.1 Gradient Times Input is Sensitive to Mean Shift of Inputs

We find that the multiplication of raw gradients by the image fails to satisfy input invariance. In Fig. 3 GI produces different saliency heatmaps for both networks.

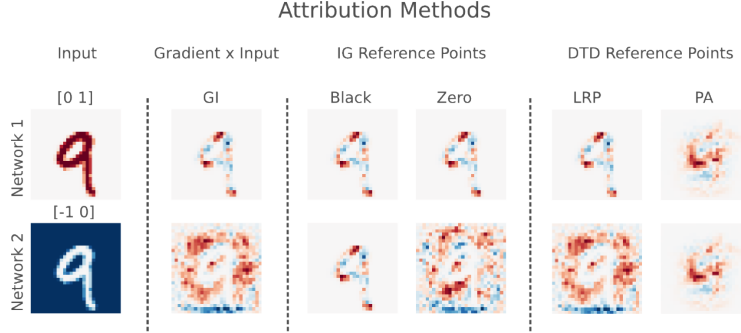


Figure 3: Evaluation of attribution method sensitivity using MNIST with a  $[0,1]$  encoding for network  $f_1$  and a  $[-1,0]$  encoding for network  $f_2$ . Gradient x Input, IG and DTD with a zero reference point, which is equivalent to LRP [1, 6], do not satisfy input invariance and produce different attributions for each network. IG with a black image reference point and DTD with a PA reference point are not sensitive to a mean shift in input.

In 3.2 we determined that a saliency heatmap of gradients gradient does satisfy input invariance. This breaks when the gradients are multiplied with the input image.

$$s_j = \frac{\partial f(\mathbf{x})_j}{\partial \mathbf{x}} \odot \mathbf{x}.$$

Multiplying by the input fails to satisfy input invariance because the input shift is carried through to final attribution. Naive multiplication by the input, as noted by [12], also constrains attribution without justification to inputs that are not 0.

### 3.3.2 Reliability of Reference Point Methods Depends Upon the Choice of Reference

Both Integrated Gradients IG, [14] and Deep Taylor Decomposition DTD, [6] determine the importance of inputs relative to a reference point. DTD refers to this as the root point and IG terms the reference point a baseline. The choice of reference point is not determined *a priori* by the method and is instead a hyperparameter of the attribution task.

The choice of reference point determines all subsequent attribution. In Fig. 1 IG and DTD show different attribution depending on the choice of reference point. We show that IG and DTD only satisfy input invariance contingent on the choice of reference point and the type of transformation considered.

**Integrated gradients** (IG) ([14]) attributes the predicted score to each input with respect to a baseline  $\mathbf{x}_0$ . This is achieved by constructing a set of inputs interpolating between the baseline and the input.

$$s_j = (\mathbf{x} - \mathbf{x}_0) \odot \int_{\alpha=0}^1 \frac{\partial f(\mathbf{x}_0 + \alpha(\mathbf{x} - \mathbf{x}_0))_i}{\partial \mathbf{x}} d\alpha$$

Since this integral cannot be computed analytically, it is approximated by a finite sum ranging over  $\alpha \in [0, 1]$ .

$$s_j = (\mathbf{x} - \mathbf{x}_0) \odot \sum_{\alpha} \frac{\partial f(\mathbf{x}_0 + \alpha(\mathbf{x} - \mathbf{x}_0))}{\partial \mathbf{x}}.$$

We evaluate whether two possible IG reference points satisfy input invariance. Firstly, we consider an image populated uniformly with the minimum pixel from the dataset ( $\mathbf{x}_0 = \min(\mathbf{x})$ ) (black image) and a zero vector image. In Fig. 3, a black image reference point produces identical attribution heatmaps whereas a zero vector reference point is not input invariant.

IG using a black image reference point is not sensitive to the constant shift in input because  $\mathbf{x}_0 = \min(\mathbf{x})$  is determined after the mean shift of the input so the difference between  $\mathbf{x}$  and  $\mathbf{x}_0$  remains

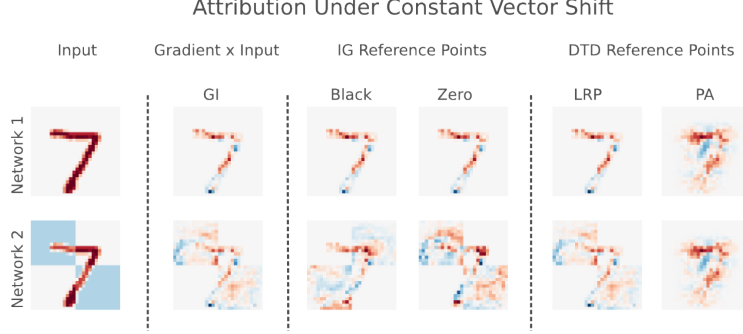


Figure 4: Evaluation of attribution method sensitivity using MNIST. Gradient x Input, all IG reference points and DTD with a LRP reference point do not satisfy input invariance and produce different attributions for each network. DTD with a PA reference point is not sensitive to the transformation of the input.

the same for both networks. In network 1 this is  $(\mathbf{x}_1) - \min(\mathbf{x}_1)$  and in network 2 this is  $(\mathbf{x}_2 + \mathbf{m}_2) - \min(\mathbf{x}_2 + \mathbf{m}_2)$ .

IG with a zero vector reference point fails to satisfy input invariance because while the difference in network 1 is  $(\mathbf{x}_1 - \mathbf{x}_0)$ , the difference in network 2 becomes  $(\mathbf{x}_2 + \mathbf{m}_2) - \mathbf{x}_0$ .

It is possible to construct a constant vector that will break the reliability of using a black image as a baseline. We consider a transformation  $\mathbf{x}_2^i$  of the input  $\mathbf{x}_1^i$  where the constant vector ( $\mathbf{m}_2$ ) added to  $\mathbf{x}_1^i$  is an image of a checkered box. Consistent with **Section 2** the relationship between  $\mathbf{x}_1^i$  and the transformed input  $\mathbf{x}_2^i$  is the addition of the checkered box image vector  $\mathbf{m}_2$ .

In Fig. 4 shows that we are able to manipulate the attribution heatmap of an MNIST prediction so that  $\hat{\mathbf{x}}$ , an image of checkered boxes, appears for all reference points except for PA. This constant vector transformation causes all IG reference points to fail to satisfy input invariance.

**Deep Taylor Decomposition (DTD)** determines attribution relative to a reference point neuron. DTD can satisfy input invariance if the right reference point is chosen. In the general formulation, the attribution of an input neuron  $j$  is initialized to be equal to the output of that neuron. The attribution of other output neurons is set to zero. This attribution is backpropagated to input neurons using the following distribution rule where  $s_j^l$  is the attribution assigned to neuron  $j$  in layer  $l$ :

$$s_j^{output} = y, \quad s_{k \neq j}^{output} = 0, \quad s^{l-1,j} = \frac{\mathbf{w} \odot (\mathbf{x} - \mathbf{x}_0)}{\mathbf{w}^T \mathbf{x}} s_j^l.$$

We evaluate the input invariance of DTD using a reference point determined by Layer-wise Relevance Propagation (LRP,[1]) and PatternAttribution (PA). In Fig. 3, DTD satisfies input invariance when using a reference point defined by PA however fails to satisfy input invariance when using a reference point defined by LRP.

LRP is sensitive to the input shift because it is a case of DTD where a zero vector is chosen as the root point.<sup>2</sup> The back-propagation rule becomes:

$$s_j^{output} = y, \quad s_{k \neq i}^{output} = 0, \quad s^{l-1,j} = \frac{\mathbf{w} \odot \mathbf{x}}{\mathbf{w}^T \mathbf{x}} s_j^l.$$

$s^{l-1,j}$  depends only upon the input and so attribution will change between network 1 and 2 because  $\mathbf{x}_1$  and  $\mathbf{x}_2$  differ by a constant vector.

PatternAttribution (PA) satisfies input invariance because the reference point  $\mathbf{x}_0$  is defined as the natural direction of variation in the data [5]. This natural direction is determined by the covariance of the data and thus compensates explicitly for the constant vector shift of the input. Therefore it is by construction input invariant.

<sup>2</sup>This case of DTD is called the *z - rule* and can be shown to be equivalent to Layer-wise Relevance Propagation [1, 6]. Under specific circumstances, LRP is also equivalent to the gradient times input [4, 9].

The PA root point is:

$$\mathbf{x}_0 = \mathbf{x} - \mathbf{a}\mathbf{w}^T \mathbf{x} \quad (1)$$

where  $\mathbf{a}^T \mathbf{w} = 1$ .

In a linear model:

$$\mathbf{a} = \frac{\text{cov}[\mathbf{x}, y]}{\mathbf{w}^T \text{cov}[\mathbf{x}, y]}. \quad (2)$$

For neurons followed by a ReLU non-linearity the vector  $\mathbf{a}$  accounts for the non-linearity and is computed as:

$$\mathbf{a} = \frac{E_+[\mathbf{x}, y] - E_+[\mathbf{x}]E[y]}{\mathbf{w}^T (E_+[\mathbf{x}, y] - E_+[\mathbf{x}]E[y])}.$$

Here  $E_+$  denotes the expectation taken over values where  $y$  is positive.

PA reduces to the following step:

$$s_i^{output} = y, \quad s_{j \neq i}^{output} = 0, \quad \mathbf{s}^{l-1, i} = \mathbf{w} \odot \mathbf{a} s_i^l.$$

The vector  $\mathbf{a}$  depends upon covariance and thus compensates the mean shift of the input. The attribution for both networks is thus identical.

## 4 The Importance of Choosing an Appropriate Reference Point

IG and DTD satisfy input invariance when certain reference points or/and input transformations are considered. The choice of reference point is also important because it determines all subsequent attribution. In fig.1 attribution visually diverges for the same method if multiple reference points are considered.

A reasonable reference point choice will naturally depend upon domain and task. For example, [14] suggests that a black image is a natural reference point for image recognition tasks whereas a zero vector is a reasonable choice for text based networks. However, we have shown that the choice of reference point can lead to very different results. Unintentional misrepresentation of the model is very possible when the implications of attribution using a given reference point are unclear. Thus far, we have discussed attribution for image recognition tasks with the assumption that pre-processing steps are known and visual inspection of the points determined to be salient is possible. For audio and language based models where visual inspection is difficult or inappropriate, identifying failure points or how attribution varies under different baselines poses a challenge.

If we cannot determine the implications of reference point choice, we are limited in our ability to say anything about the reliability of the method. To demonstrate this point, we construct a constant shift of the input that takes advantage of the input invariance points of failure we have already identified.

In the following experiment, we construct a constant vector shift using a hand drawn image of cat. Network 1 is the same as introduced in **Section 2**. The raw image can be seen in Fig. 5. Consistent with **Section 2** the relationship between  $\mathbf{x}_1^i$  and the transformed input  $\mathbf{x}_2^i$  is the addition of a constant vectors  $\mathbf{m}_2$ .

$$\forall i, \mathbf{x}_2^i = \mathbf{x}_1^i + \mathbf{m}_2.$$

We construct  $\mathbf{m}_2$  by choosing a desired attribution  $\hat{s}$  that should be assigned to a specific sample  $\hat{\mathbf{x}}$  when the gradient is multiplied with the input.

$\mathbf{m}_2$  is constructed to ensure that the specific  $\mathbf{x}_2^i$  receives the desired attribution as follows:

$$\mathbf{m}_2 = \frac{\hat{s}}{\frac{\partial f_1(\hat{\mathbf{x}})}{\partial \mathbf{x}}} - \hat{\mathbf{x}}.$$

We clip the shift to be within [-.3,.3] so that the MNIST digit is still visible, if we do not clip the end attribution would only show the cat.

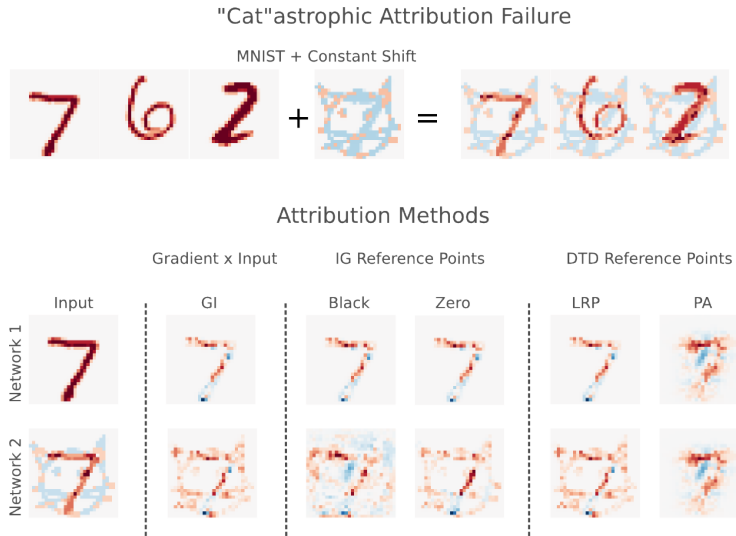


Figure 5: Evaluation of attribution method sensitivity using MNIST. Gradient x Input, IG with both a black and zero reference point and DTD with a LRP reference point, do not satisfy input invariance and produce different attribution for each network. DTD with a PA reference point is not sensitive to the transformation of the input.

In Fig. 5 transforming the input in this manner allows purposeful misrepresentation of the attribution. All methods, except for PA, fail to satisfy input invariance and visibly show a cat as the explanation for an MNIST prediction.

How can we avoid breaks in input invariance? PA is invariant to the input transformations considered because it relies on the covariance of the data which compensates for the shift. If the data had been normalized prior to attribution many of the methods considered would still satisfy input invariance. However, this is far from a systematic treatment of the reference point selection as there are input transformations outside of our experiment scope where this would not be sufficient. We believe an important research agenda is furthering the understanding of reference point choice that guarantee reliability without relying on case-by-case solutions.

## 5 Conclusion

Saliency methods are powerful tools to gain intuition about our model. We show that numerous methods fail to attribute correctly when a constant vector shift is applied to the input. More worryingly, we show that we are able to purposefully create a deceptive explanation of the network using a hand drawn cat image.

We introduce *input invariance* as a prerequisite for reliable attribution. Our treatment of input invariance is restricted to demonstrating that there is at least one input transformation (a constant vector shift to the input) that causes numerous saliency methods to attribute incorrectly. This work is motivated by our belief that saliency methods remain valuable tools to gain intuition about the network. Understanding where they fail equips researchers with the tools to appropriately weigh the explanations these models provide.

Guaranteeing the reliability of saliency methods is crucial in tasks where visual inspection of results is not easy or the costs of incorrect attribution is high. For example, human inspection of the attribution for an image recognition task would catch the cat attack experiment (described in section 4). However, it is unclear how we would catch the same purposeful manipulation or an unintentional misrepresentation in a language or audio model where inspection is not possible or opaque.

Determining how saliency methods fail is an important stepping stone to understanding where and how we should use these methods. An urgent research agenda, and a requirement for the use of deep neural networks in domains like medicine, is evaluating which methods and/or reference points consistently guarantee reliability for all possible transformations.



## Acknowledgements

We would like to acknowledge the thoughtful feedback and guidance of Gregoir Montavon, Mukund Sundararajan, Ankur Taly, Douglas Eck and Jonas Kemp.

## References

- [1] 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*, 10(7):e0130140, 2015.
- [2] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.
- [3] S. Haufe, F. Meinecke, K. Görgen, S. Dähne, J.-D. Haynes, B. Blankertz, and F. Bießmann. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage*, 87:96–110, 2014.
- [4] P.-J. Kindermans, K. Schütt, K.-R. Müller, and S. Dähne. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*, 2016.
- [5] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, and S. Dähne. Learning how to explain neural networks: Patternnet and patternattribution. *arXiv preprint arXiv:1705.05598v2*, 2017.
- [6] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.
- [7] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016.
- [8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [9] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences. *CoRR*, abs/1605.01713, 2016.
- [10] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*, 2014.
- [11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [12] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
- [13] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR*, 2015.
- [14] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.
- [15] J. Yosinski, J. Clune, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. In *ICML Workshop on Deep Learning*, 2015.
- [16] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.
- [17] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017.