Fairwashing Explanations with Off-Manifold Detergent

Christopher J. Anders [1] **Plamen Pasliev** [1] **Ann-Kathrin Dombrowski** [1] **Klaus-Robert Müller** [1 2 3] **Pan Kessel** [1]

Abstract

Explanation methods promise to make black-box classifiers more transparent. As a result, it is hoped that they can act as proof for a sensible, fair and trustworthy decision-making process of the algorithm and thereby increase its acceptance by the end-users. In this paper, we show both theoretically and experimentally that these hopes are presently unfounded. Specifically, we show that, for any classifier g, one can always construct another classifier $\tilde{g}$ which has the same behavior on the data (same train, validation, and test error) but has arbitrarily manipulated explanation maps. We derive this statement theoretically using differential geometry and demonstrate it experimentally for various explanation methods, architectures, and datasets. Motivated by our theoretical insights, we then propose a modification of existing explanation methods which makes them significantly more robust.

1. Introduction

Explanation methods[4] are increasingly adopted by machine learning practitioners and incorporated into standard deep learning libraries (Kokhlikyan et al., 2019; Alber et al., 2019; Ancona et al., 2018). The interest in explainability is partly driven by the hope that explanations can act as proof for a sensible, fair, and trustworthy decision-making process(Aïvodji et al., 2019; Lapuschkin et al., 2019). As an example, a bank could provide explanations for its rejection of a loan application. By doing so, the bank can demonstrate that the decision was not based on illegal or ethically questionable features. It can furthermore provide feedback to the customer. In some situations, an explanation of an algorithmic decision may even be required by law.

However, this hope is based on the assumption that explanations faithfully reflect the underlying mechanisms of the algorithmic decision. In this work, we demonstrate unequivocally that this assumption should not be made carelessly because explanations can be easily manipulated.

In more detail, we show theoretically that for any classifier g, one can always find another classifier $\tilde{g}$ which agrees with the original g on the entire data manifold but has (almost) completely controlled explanations. This surprising result is established using techniques of differential geometry. We then demonstrate experimentally that one can easily construct such manipulated classifiers $\tilde{g}$.

In the example above, a bank could use a manipulated classifier $\tilde{g}$ that uses mainly unethical features, such as the gender of the applicant, but has explanations which suggest that the decision was only based on financial features.

Briefly put, the manipulability of explanations arises from the fact that the data manifold is typically low-dimensional compared to its high-dimensional embedding space. The training process only determines the classifier in directions along the manifold. However, many explanation methods are mainly sensitive to directions orthogonal to the data manifold. Since these directions are undetermined by training, they can be changed at will.

This theoretical insight allows us to propose a modification to explanation methods which make them significantly more robust with respect to such manipulations. Namely, the explanation is projected along tangential directions of the data manifold. We show, both theoretically and experimentally, that these tangent-space-projected (tsp) explanations are indeed significantly more robust. We thereby establish a novel and exciting connection between the fields of explainability and manifold learning.

In summary, our main contributions are as follows:

- Using differential geometry, we establish theoretically that popular explanation methods can be easily manipulated.

[1] Machine Learning Group, Technische Universität Berlin, Germany [2] Max-Planck-Institut für Informatik, Saarbrücken, Germany [3] Department of Brain and Cognitive Engineering, Korea University, Seoul, Korea. Correspondence to: Pan Kessel <pan.kessel@tu-berlin.de>, Klaus-Robert Müller <klaus-robert.mueller@tu-berlin.de>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

[4] See (Samek et al., 2019) and references therein for a detailed overview.

- We validate our theoretical predictions in detailed experiments for various explanation methods, classifier architectures, and datasets, as well as for different tasks.
- We propose a modification to existing explanation methods which make them more robust with respect to these manipulations.
- In doing so, we relate explainability to manifold learning.

1.1. Related Works

This work was crucially inspired by (Heo et al., 2019). In this reference, adversarial model manipulation for explanations is proposed. Specifically, the authors empirically show that one can train models such that they have structurally different explanations while suffering only a very mild drop in classification accuracy compared to their unmanipulated counterparts. For example, the adversarial model manipulation can change the positions of the most relevant pixels in each image or increase the overall sum of relevances in a certain subregion of the images. Contrary to their work, we analyze this problem theoretically. Our analysis leads us to demonstrate a stronger form of manipulability. Namely, the model can be manipulated such that it structurally reproduces arbitrary target explanations while keeping all class probabilities the same for all data points. Our theoretical insights not only illuminate the underlying reasons for the manipulability but also allow us to develop modifications of existing explanation methods which make them more robust. Another approach (Kindermans et al., 2019) adds a constant shift to the input image, which is then eliminated by changing the bias of the first layer. For some methods, this leads to a change in the explanation map. Contrary to our approach, this requires a shift in the data. In (Adebayo et al., 2018), explanation maps are changed by randomization of (some of) the network weights. This is different to our method as it dramatically changes the output of the network and is proposed as a consistency check of explanations. In (Dombrowski et al., 2019) and (Ghorbani et al., 2019), it is shown that explanations can be manipulated by an infinitesimal change in input while the output of the network is approximately unchanged. Contrary to this approach, we manipulate the model and keep the input unchanged.

1.2. Explanation Methods

We consider a classifier $g : \mathbb{R}^D \to \mathbb{R}^K$ which classifies an input $x \in \mathbb{R}^D$ in K categories with the predicted class given by $k = \arg\max_i g(x)_i$. The explanation method is denoted by $h_g : \mathbb{R}^D \to \mathbb{R}^D$ and associates an input x with an explanation map $h_g(x)$ whose components encode the relevance score of each input for the classifier's prediction. We note that, by convention, explanation maps are usually calculated with respect to the classifier before applying the final softmax non-linearity (Kokhlikyan et al., 2019; Alber et al., 2019; Ancona et al., 2018). Throughout the paper, we will therefore denote this function as g.

We use the following explanation methods:

Gradient: The map $h_g(x) = \frac{\partial g}{\partial x}(x)$ is used and quantifies how infinitesimal perturbations in each pixel change the prediction $g(x)$ (Simonyan et al., 2014; Baehrens et al., 2010).

x $\odot$ Grad: This method uses the map $h_g(x) = x \odot \frac{\partial g}{\partial x}(x)$ (Shrikumar et al., 2017). For linear models, the exact contribution of each pixel to the prediction is obtained.

Integrated Gradients: This method defines

$$h_g(x) = (x - \bar{x}) \odot \int_0^1 \frac{\partial g(\bar{x} + t(x - \bar{x}))}{\partial x} \mathrm{d}t$$

where $\bar{x}$ is a suitable baseline. We refer to the original reference (Sundararajan et al., 2017) for more details.

Layer-wise Relevance Propagation (LRP): This method (Bach et al., 2015; Montavon et al., 2017) propagates relevance backwards through the network. In our experiments, we use the following setup: for the output layer, relevance is given by

$$R_i^L = \delta_{i,k} = \begin{cases} 1, & \text{for } i = k \\ 0, & \text{for } i \neq k \end{cases},$$

which is then propagated backwards through all layers but the first using the z^+-rule

$$R_i^l = \sum_j \frac{x_i^l (W^l)_{ji}^+}{\sum_i x_i^l (W^l)_{ji}^+ + \epsilon} R_j^{l+1}, \qquad (1)$$

where $(W^l)^+$ denotes the positive weights of the l-th layer, x^l is the activation vector of the l-th layer, and $\epsilon > 0$ is a small constant ensuring numerical stability. For the first layer, we use the $z^{\mathcal{B}}$-rule to account for the bounded input domain

$$R_i^0 = \sum_j \frac{x_j^0 W_{ji}^0 - l_j (W^0)_{ji}^+ - h_j (W^0)_{ji}^-}{\sum_i (x_j^0 W_{ji}^0 - l_j (W^0)_{ji}^+ - h_j (W^0)_{ji}^-)} R_j^1,$$

where l_i and h_i are the lower and upper bounds of the input domain respectively.
For theoretical analysis, we consider the ϵ-rule in all layers for simplicity. This rule is obtained by substituting $(W^l)^+ \to W^l$ in (1). We refer to the resulting method as ϵ-LRP.

This choice of methods is necessarily not exhaustive. However, it covers two classes of attribution methods, i.e. propagation and gradient-based explanations. Furthermore, the chosen methods are widely used in practice (Kokhlikyan et al., 2019; Alber et al., 2019; Ancona et al., 2018).

2. Manipulation of Explanations

In this section, we will theoretically deduce that explanation methods can be arbitrarily manipulated by adversarially training a model.

2.1. Mathematical Background

In the following, we will briefly summarize the basic tools of differential geometry before applying them in the context of explainability in the next section. For additional technical details, we refer to Appendix A.1.

A D-dimensional manifold M is a topological space which locally resembles $\mathbb{R}^D$. More precisely, for each $p \in M$, there exists a subset $U \subset M$ containing p and a diffeomorphism $\phi : U \to \tilde{U} \subset \mathbb{R}^D$. The pair (U, ϕ) is called *coordinate chart* and the component functions x^i of $\phi(p) = (x^1(p), \ldots, x^D(p))$ are called *coordinates*.

A d-dimensional submanifold S is a subset of M which is itself a d-dimensional manifold. M is called the embedding manifold of S. A *properly embedded submanifold* $S \subset M$ is a submanifold embedded in M which is also closed as a set.

Let $p \in M$ be a point on a manifold M and $\gamma : \mathbb{R} \to M$ with $\gamma(0) = p$ a curve through the point p. The set of tangent vectors $d\gamma = \frac{d}{dt}\gamma(t)|_{t=0}$ of all curves through p forms a vector space of dimension D. This vector space is known as *tangent space* T_pM. Let (U, ϕ) be a coordinate chart on M with coordinates x. We can then define $\phi \circ \lambda_k(t) = (x^1(p), \ldots, x^k(p) + t, \ldots, x^D(p))$ with $k \in \{1, \ldots, D\}$. This implicitly defines curves $\lambda_k : \mathbb{R} \to M$ through p. We denote the corresponding tangent vectors as $\partial_k := \frac{d}{dt}\lambda_k(t)|_{t=0}$ and it can be shown that they form a basis of the tangent space T_pM.

A *vector field* V on M associates with every point $x \in M$ an element of the corresponding tangent space, i.e. $V(x) \in T_xM$.[5] A conservative vector field V is a vector field that is the gradient of a function $f : M \to \mathbb{R}$, i.e. $V(x) = \nabla f(x)$. For submanifolds S, there are two different notions of vector fields. A vector field V *on* the submanifold S associates to every point on S a vector in its corresponding tangent space T_xS, i.e. $V(x) \in T_xS$. A vector field V *along* the submanifold S associates to every point on S a vector in the corresponding tangent space of the embedding manifold M, i.e. $V(x) \in T_xM$. These concepts can be related as follows: the tangent space T_xM can be decomposed into the tangent space T_xS of S and its orthogonal complement $T_xS^\perp$, i.e. $T_xM = T_xS \oplus T_xS^\perp$. A vector field along S which only takes values in the first summand T_xS is also a vector field on S.

With these definitions, we can now state a crucial theorem for our theoretical analysis. In Appendix A.1, we show that:

Theorem 1 *Let $S \subset M$ be d-dimensional submanifold properly embedded in the D-dimensional manifold M. Let $V = \sum_{i=d+1}^{D} v^i \partial_i$ be a conservative vector field along S which assigns a vector in $T_pS^\perp$ for each $p \in S$. For any smooth function $f : S \to \mathbb{R}$, there exists a smooth extension $F : M \to \mathbb{R}$ such that*

$$F|_S = f$$

where $F|_S$ denotes the restriction of F on the submanifold S. Furthermore, the derivative of the extension F is given by

$$\nabla F(x) = (\nabla_1 f(x), \ldots \nabla_d f(x), v^{d+1}(x), \ldots, v^D(x))$$

for all $x \in S$.

Technical details not withstanding, this theorem states that a function f defined on a submanifold S can be extended to the entire embedding manifold M. The extension's derivatives orthogonal to the submanifold S can be freely chosen.

This theorem is a generalization of the well-known submanifold extension lemma (see, for example, Lemma 5.34 in (Lee, 2012)) in that it not only shows that an extension exists but also that one has control over the gradient of the extension F. While we could not find such a statement in the literature, we suspect that it is entirely obvious to differential geometers but typically not needed for their purposes.

2.2. Explanation Manipulation: Theory

From Theorem 1, it follows under a mild assumption that one can always construct a model $\tilde{g}$ such that it closely reproduces arbitrary target explanations but has the same training, validation, and test loss as the original model g.

Assumption: the data lies on a d-dimensional submanifold $S \subset M$ properly embedded in the manifold $M = \mathbb{R}^D$. The data manifold S is of much lower dimensionality than its embedding space M, i.e.

$$\epsilon \equiv \frac{d}{D} \ll 1 \,. \tag{2}$$

We stress that this assumption is also known as the manifold conjecture and is expected to hold across a wide range of machine learning tasks. We refer to (Goodfellow et al., 2016) for a detailed discussion.

Under this assumption, the following theorem can be derived for the Gradient, $x \odot$ Grad, and ϵ-LRP methods (only the proof for the Gradient method is given; see Appendix 2 for other methods):

[5] More rigorously, vector fields are defined in terms of the tangent bundle. We refrain from introducing bundles for accessibility.

Theorem 2 *Let $h_g : \mathbb{R}^D \to \mathbb{R}^D$ be the explanation of classifier $g : \mathbb{R}^D \to \mathbb{R}$ with bounded derivatives $|\nabla_i g(x)| \leq C \in \mathbb{R}_+$ for $i = 1, \dots, D$.*

For a given target explanation $h^t : \mathbb{R}^D \to \mathbb{R}^D$, there exists another classifier $\tilde{g} : \mathbb{R}^D \to \mathbb{R}$ which completely agrees with the classifier g on the data manifold S, i.e.

$$\tilde{g}|_S = g|_S \,. \tag{3}$$

In particular, both classifiers have the same *train, validation, and test loss.*

However, its explanation $h_{\tilde{g}}$ closely resembles the target h^t, i.e.

$$MSE(h_{\tilde{g}}(x), h^t(x)) \leq \epsilon \qquad \forall x \in S\,, \tag{4}$$

where $MSE(h, h') = \frac{1}{D}\sum_{i=1}^{D}(h_i - h'_i)^2$ denotes the mean-squared error and $\epsilon = \frac{d}{D}$.

Proof: By Theorem 1, we can find a function G which agrees with g on the data manifold S but has the derivative

$$\nabla G(x) = (\nabla_1 g(x), \dots \nabla_d g(x), h^t_{d+1}(x), \dots, h^t_D(x))$$

for all $x \in S$. By definition, this is its gradient explanation $h_G = \nabla G$.

As explained in Appendix A.2.1, we can assume without loss of generality that $|\nabla_i g(x)| \leq 0.5$ for $i \in \{1, \dots, D\}$. We can furthermore rescale the target map such that $|h^t_i| \leq 0.5$ for $i \in \{1, \dots, D\}$. This rescaling is merely conventional as it does not change the relative importance h_i of any input component x_i with respect to the others. It then follows that

$$\text{MSE}(h_G(x), h^t(x)) = \tfrac{1}{D}\sum_{i=1}^{D}(\nabla_i G(x) - h^t_i(x))^2 \,.$$

This sum can be decomposed as

$$\tfrac{1}{D}\sum_{i=1}^{d}\underbrace{\left(\nabla_i g(x) - h^t_i(x)\right)^2}_{\leq 1} + \tfrac{1}{D}\sum_{i=d+1}^{D}\underbrace{\left(\nabla_i G(x) - h^t_i(x)\right)^2}_{=0}$$

and from this, it follows that

$$\text{MSE}(h_G(x), h^t(x)) \leq \frac{d}{D} = \epsilon\,,$$

The proof then concludes by identifying $\tilde{g} = G$. □

Intuition: Somewhat roughly, this theorem can be understood as follows: two models, which behave identically on the data, need to only agree on the low-dimensional submanifold S. The gradients "orthogonal" to the submanifold S are completely undetermined by this requirement. By the manifold assumption, there are however much more "orthogonal" than "parallel" directions and therefore the explanation is largely controlled by these. We can use this fact to closely reproduce an arbitrary target while keeping the function's values on the data unchanged.

We stress however that there are a number of non-trivial differential geometric arguments needed in order to make these statements rigorous and quantitative. For example, it is entirely non-trivial that an extension to the embedding manifold exists for arbitrary choice of target explanation. This is shown by Theorem 1 whose proof is based on a differential geometric technique called partition of the unity subordinate to an open cover. See Appendix A.1 for details.

2.3. Explanation Manipulation: Methods

Flat Submanifolds and Logistic Regression: The previous theorem assumes that the data lies on an arbitrarily curved submanifold and therefore has to rely on relatively involved mathematical concepts of differential geometry. We will now illustrate the basic ideas in a much simpler context: we will assume that the data lies on a d-dimensional flat hyperplane $S \subset \mathbb{R}^D$.[6] The points on the hyperplane S obey the relation

$$\forall x \in S: \quad (\hat{w}^{(i)})^T x = b_i\,, \quad i \in \{1, \dots, D-d\}\,, \tag{5}$$

where $\{\hat{w}^{(i)} \in \mathbb{R}^D \mid i = 1, \dots, D-d\}$ are a set of normal vectors to the hyperplane S and $b_i \in \mathbb{R}$ are the affine translations. We furthermore assume that we use logistic regression as the classification algorithm, i.e.

$$g(x) = \sigma(w^T x + c)\,, \tag{6}$$

where $w \in \mathbb{R}^D$, $c \in \mathbb{R}$ are the weights and the bias respectively and $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function. This classifier has the gradient explanation[7]

$$h_{\text{grad}}(x) = w\,, \tag{7}$$

We can now define a modified classifier by

$$\tilde{g}(x) = \sigma\left(w^T x + \sum_i \lambda_i ({\hat{w}^{(i)}}^T x - b_i) + c\right)\,, \tag{8}$$

for arbitrary $\lambda_i \in \mathbb{R}$. By (5), it follows that both classifiers agree on the data manifold S, i.e.

$$\forall x \in S: \qquad g(x) = \tilde{g}(x)\,, \tag{9}$$

and therefore have the same train, validation, and test error. However, the gradient explanations are now given by

$$h_{\text{grad}}(x) = w + \sum_i \lambda_i \hat{w}^{(i)}\,. \tag{10}$$

[6] In mathematics, these submanifolds are usually referred to as d-flats and only the case $d = D - 1$ is called hyperplane. We refrain from this terminology.

[7] We recall that in calculating the explanation map, we take the derivative *before* applying the final activation function.

Since the λ_i can be chosen freely, we can modify the explanations arbitrarily in directions orthogonal to the data submanifold S (parameterized by the normal vectors $\hat{w}^{(i)}$). Similar statements can be shown for other explanation methods and we refer to the Appendix A.3 for more details.

As we will discuss in Section 2.4, one can use these tricks even for data which does not (initially) lie on a hyperplane.

General Case: For the case of arbitrary neural networks and curved data manifolds, we cannot analytically construct the manipulated model $\tilde{g}$. We therefore approximately obtain the model $\tilde{g}$ corresponding to the original model g by minimizing the loss

$$\mathcal{L} = \sum_{x_i \in \mathcal{T}} ||g(x_i) - \tilde{g}(x_i)||^2 + \gamma \sum_{x_i \in \mathcal{T}} ||h_{\tilde{g}}(x_i) - h^t||^2 \,, \tag{11}$$

by stochastic gradient descent with respect to the parameters of $\tilde{g}$. The training set is denoted by $\mathcal{T}$ and $h^t \in \mathbb{R}^D$ is a specified target explanation. Note that we could also use different targets for various subsets of the data but we will not make this explicit to avoid cluttered notation. The first term in the loss $\mathcal{L}$ ensures that the models g and $\tilde{g}$ have approximately the same output while the second term encourages the explanations of $\tilde{g}$ to closely reproduce the target h^t. The relative weighting of these two terms is determined by the hyperparameter $\gamma \in \mathbb{R}_+$.

As we will demonstrate experimentally, the resulting $\tilde{g}$ will closely reproduce the target explanation h^t and have (approximately) the same output as g. Crucially, both statements will be seen to hold also for the test set.

2.4. Explanation Manipulation: Practice

In this section, we will demonstrate manipulation of explanations experimentally. We will first discuss applying logistic regression to credit assessment and then proceed to the case of deep neural networks in the context of image classification. The code for all our experiments is publicly available at https://github.com/fairwashing/fairwashing.

Credit Assessment: In the following, we will suppose that a bank uses a logistic regression algorithm to classify whether a prospective client should receive a loan or not. The classification uses the features $x = (x_{\text{gender}}, x_{\text{income}})$ where

$$x_{\text{gender}} = \begin{cases} 1, & \text{for male} \\ -1, & \text{for female} \end{cases} \tag{12}$$

and x_{income} is the income of the applicant. Normalization is chosen such that the features are of the same order of magnitude. Details can be found in the Appendix B.

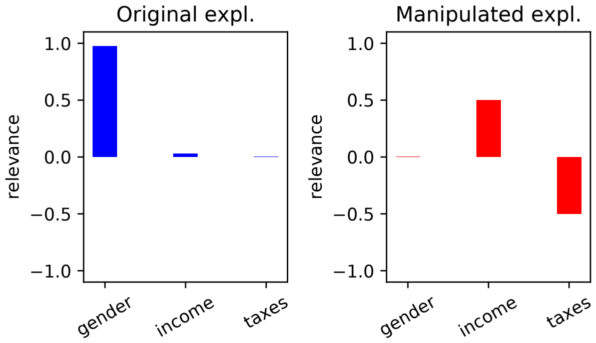

Figure 1. x⊙Grad explanations for **original classifier g** and **manipulated $\tilde{g}$** highlight completely different features. Colored bars show the median of the explanations over multiple examples.

We then define a logistic regression classifier g by choosing the weights $w = (0.9, 0.1)$, i.e. female applicants are severely discriminated against. The discriminating nature of the algorithm may be detected by inspecting, for example, the gradient explanation maps $h^{\text{grad}}_g = w$.

Conversely, if the explanations did not show any sign of discrimination for another classifier $\tilde{g}$, the user may interpret this as a sign of its trustworthiness and fairness.

However, the bank can easily ”fairwash” the explanations, i.e. hide the fact that the classifier is sexist. This can be done by adding new features which are linearly dependent on the previously used features. As a simple example, one could add the applicant's paid taxes x_{taxes} as a feature. By definition, it holds that

$$x_{\text{taxes}} = 0.4\, x_{\text{income}} \,, \tag{13}$$

where we assume that there is a fixed tax rate of 0.4 on all income. The features used by the classifier are now $x = (x_{\text{gender}}, x_{\text{income}}, x_{\text{taxes}})$. By (13), all data samples x obey

$$\hat{w}^T x = 0 \qquad \text{with} \qquad \hat{w} = (0, 0.4, -1)\,. \tag{14}$$

Therefore, the original classifier $g(x) = \sigma(w^T x)$ with $w = (0.9, 0.1, 0)$ leads to the same output as the classifier $\tilde{g}(x) = \sigma(w^T x + 1000\, \hat{w}^T x)$. However, as shown in Figure 1, the classifier $\tilde{g}$ has explanations which suggest that the two financial features (and *not* the applicant's gender) are important for the classification result.

This example is merely an (oversimplified) illustration of a general concept: for each additional feature which linearly depends on the previously used features, a condition of the form (14) for some normal vector $\hat{w}$ is obtained. We can then construct a classifier with arbitrary explanation along each of these normal vectors.

Image Classification: We will now experimentally demonstrate the practical applicability of our methods in the context of image classification with deep neural networks.

Datasets: We consider the MNIST, FashionMNIST, and CIFAR10 datasets. We use the standard training and test sets for our analysis. The data is normalized such that it has mean zero and standard deviation one. We sum the explanations over the absolute values of its channels to get the relevance per pixel. The resulting relevances are then normalized to have a sum of one.

Models: For CIFAR10, we use the VGG16 (Simonyan & Zisserman, 2015) architecture. For FashionMNIST and MNIST, we use a four layer convolutional neural network. We train the model g by minimizing the standard cross entropy loss for classification. The manipulated model $\tilde{g}$ is then trained by minimizing the loss (11) for a given target explanation h^t. This target was chosen to have the shape of the number 42. For more details about the architectures and training, we refer to the Appendix D.

Quantitative Measures: We assess the similarity between explanation maps using three quantitative measures: the structural similarity index (SSIM), the Pearson correlation coefficient (PCC) and the mean squared error (MSE). SSIM and PCC are relative similarity measures with values in $[0, 1]$, where larger values indicate high similarity. The MSE is an absolute error measure for which values close to zero indicate high similarity. We also use the MSE metric as well as the Kullback-Leibler divergence for assessing similarity of the class scores of the manipulated model $\tilde{g}$ and the original network g.

Results: For all considered models, datasets, and explanation methods, we find that the manipulated model $\tilde{g}$ has explanations which closely resemble the target map h^t, e.g. the SSIM between the target and manipulated explanations is of the order 0.8. At the same time, the manipulated network $\tilde{g}$ has approximately the same output as the original model g, i.e. the mean-squared error of the outputs after the final softmax non-linearity is of the order 10^{-3}. The classification accuracy is changed by about 0.2 percent.

Figure 2 illustrates this for examples from the FashionMNIST and CIFAR10 test sets. We stress that we use a single model for Gradient, x⊙Grad, and Integrated Gradient methods which demonstrates that the manipulation generalizes over all considered gradient-based methods.

The left-hand-side of Figure 3 shows quantitatively that manipulated model $\tilde{g}$ closely reproduces the target map h^t over the entire test set of FashionMNIST. We refer to the Appendix D for additional similarity measures, examples, and quantitative analysis for all datasets.

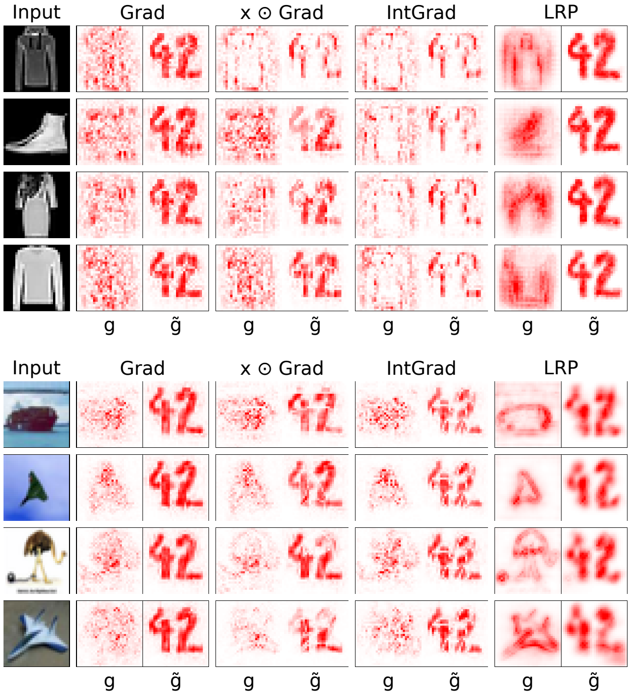

Figure 2. Example explanations from the original model g (left) and the manipulated model $\tilde{g}$ (right). Images from the test sets of FashionMNIST (**top**) and CIFAR10 (**bottom**).

3. Robust Explanations

Having demonstrated both theoretically and experimentally that explanations are highly vulnerable to model manipulation, we will now use our theoretical insights to propose explanation methods which are significantly more robust under such manipulations.

3.1. TSP Explanations: Theory

In this section, we will define a robuster *gradient* explanation method. Appendix C discusses analogous definitions for other methods.

We can formally define an explanation field H_g which associates to every point x on the data manifold S the corresponding gradient explanation $h_g(x)$ of the classifier g. We note that H_g is generically a vector field *along* the manifold since $h_g(x) \in \mathbb{R}^D \cong T_x M$, i.e. it is an element of the tangent space $T_x M$ of the embedding manifold M and *not* an element of the tangent space $T_x S$ of data manifold S.

As explained in Section 2.1, we can decompose the tangent space $T_p M$ of the embedding manifold M as follows $T_x M = T_x S \oplus T_x S^{\perp}$. Let $P : T_x M \to T_x S$ be the projection on the first summand of this decomposition. We stress that the form of the projector P depends on the point $x \in S$ but we do not make this explicit in order to simplify notation. We can then define:

Definition 1 *The tangent-space-projected (tsp) explanation field $\hat{H}_g$ is a vector field on the data manifold S. It associates to each $x \in S$, the tangent-space-projected (tsp) explanation $\hat{h}_g(x)$ given by*

$$\hat{h}_g(x) = (P \circ h_g)(x) \in T_x S \,. \tag{15}$$

Intuitively, the tsp-explanation $\hat{h}_g(x)$ is the explanation of the model g projected on the ”tangential directions” of the data manifold.

We recall from our discussion of Theorem 2 that we can always find classifiers $\tilde{g}$ which coincide with the original classifier g on the data manifold S but may differ in the gradient components orthogonal to the data manifold, i.e. for some $x \in S$ it holds that

$$(1 - P)\, \nabla g(x) \neq (1 - P)\, \nabla \tilde{g}(x) \,.$$

On the other hand, the components tangential to the manifold S agree

$$P\, \nabla g(x) = P\, \nabla \tilde{g}(x) \,, \qquad \forall x \in S \,.$$

In other words, the tsp-gradient explanations of the original model g and any such model $\tilde{g}$ are identical:

$$\hat{h}_g(x) = \hat{h}_{\tilde{g}}(x) \qquad \forall x \in S \,. \tag{16}$$

It can therefore be expected that tsp-explanations $\hat{h}_g$ are significantly more robust compared to their unprojected counterparts h_g.

For other explanation methods, the corresponding tsp-explanations may be obtained using a slightly modified projector P. We refer to Appendix C for more details.

3.2. TSP Explanations: Methods

Flat Submanifolds and Logistic Regression: Recall from Section 2.3 that for a logistic regression model $g(x) = \sigma(w^T x + c)$ with gradient explanation $h_g^{\text{grad}} = w$, we can define a manipulated model

$$\tilde{g}(x) = \sigma\left(w^T x + \sum_i \lambda_i (\hat{w}^{(i)^T} x - b_i) + c\right)$$

with gradient explanation $h_{\tilde{g}}^{\text{grad}} = w + \sum_i \lambda_i \hat{w}^{(i)}$ for arbitrary $\lambda_i \in \mathbb{R}$. Since the vectors $\hat{w}^i$ are normal to the data hypersurface S, it holds that $P\hat{w}_i = 0$. As a result, the gradient tsp-explanations of the original model g and its manipulated counterpart $\tilde{g}$ are identical, i.e.

$$\hat{h}_g^{\text{grad}} = \hat{h}_{\tilde{g}}^{\text{grad}} = Pw \,. \tag{17}$$

We discuss the case of other explanation methods in the Appendix C.1.

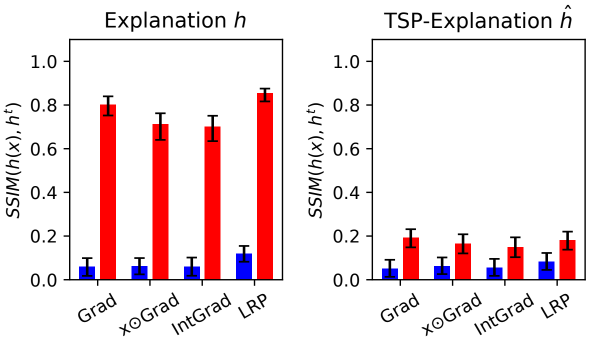

Figure 3. **Left:** SSIM of the target map h^t and explanations of **original model** g and **manipulated** $\tilde{g}$ respectively. Clearly, the manipulated model $\tilde{g}$ has explanations which closely resemble the target map h^t over the entire FashionMNIST test set. **Right:** Same as on the left but for *tsp-explanations*. The model $\tilde{g}$ was trained to manipulate the tsp-explanation. Evidently, tsp-explanations are considerably more robust than their unprojected counterparts on the left. Colored bars show the median. Errors denote the 25th and 75th percentile. Other similarity measures show similar behaviour and can be found in Appendix D.

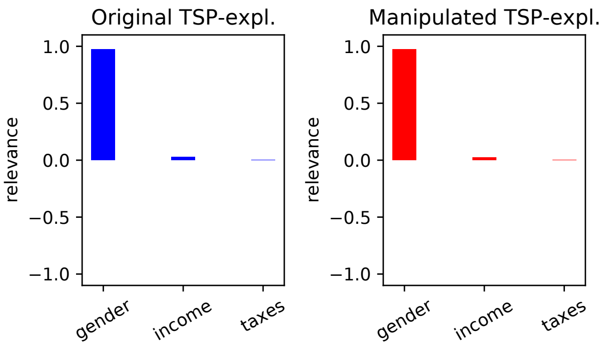

Figure 4. x⊙Grad *tsp-explanations* for **original classifier** g and **manipulated** $\tilde{g}$ highlight the same features. Colored bars show the median of the explanations over multiple examples.

General Case: In many practical applications, we do not know the explicit form of the projection matrix P. In these situations, we propose to construct P by one of the following two methods:

Hyperplane method: for a given datapoint $x \in S$, we find its k-nearest neighbours $x_1, \ldots, x_k$ in the training set. We then estimate the data tangent space $T_x S$ by constructing the d-dimensional hyperplane with minimal Euclidean distance to the points $x, x_1, \ldots, x_k$. Let this hyperplane be spanned by an orthonormal basis $q_1, \ldots q_d \in \mathbb{R}^D$. The projection matrix P on this hyperplane is then given by

$$P = \sum_{i=1}^{d} q_i \, q_i^T \, .$$

Autoencoder method: the hyperplane method requires that the data manifold is sufficiently densely sampled, i.e. the nearest neighbors are small deformations of the data point itself. In order to estimate tangent space for datasets without this property, we use techniques from the well-established field of manifold learning. Following (Shao et al., 2018), we train an autoencoder on the dataset and then perform an SVD decomposition of the Jacobian of decoder D,

$$\frac{\partial D}{\partial z} = U \, \Sigma \, V \, . \tag{18}$$

The projector is constructed from the left-singular values $u_1, \dots, u_d \in \mathbb{R}^D$ corresponding to the d largest singular values. The projector is obtained by

$$P = \sum_{i=1}^{d} u_i \, u_i^T \, . \tag{19}$$

The underlying motivation for this procedure is reviewed in Appendix C.2.

After one of these methods is used to estimate the projector P for a given $x \in S$, the corresponding tsp-explanation can be easily computed by $\hat{h}(x) = P \, h(x)$.

3.3. TSP Explanations: Practice

In this section, we will apply tsp-explanations to the examples of Section 2.4 and show that they are significantly more robust under model manipulations.

Credit Assessment: From the arguments of the previous section, it follows that the explanations of the manipulated and original model agree. We indeed confirm this experimentally, see Figure 4. We refer to the Appendix B for more details.

Image Classification: For MNIST and FashionMNIST, we use the hyperplane method to estimate the tangent space. For CIFAR10, we find that the manifold is not densely sampled enough and we therefore use the autoencoder method. This is computationally expensive and takes about 48h using four Tesla P100 GPUs. We refer to Appendix D for more details.

Figure 5 shows the tsp-explanations for the examples of Figure 2. The explanation maps of the original and manipulated model show a high degree of visual similarity. This suggests the manipulation occurred mainly in directions orthogonal to the data manifold (as the tsp-explanations are obtained from the original explanations by projecting out the corresponding components). This is also confirmed quantitatively, see Appendix D. Furthermore, tsp-explanations tend to be considerably less noisy than their unprojected counterparts (see Figure 5 vs 2). This is expected from our theoretical analysis: consider gradient explanations for concreteness. Their components orthogonal to the data manifold are undetermined by training and are therefore essentially chosen at random. This fitting noise is projected out in the tsp-explanation which results in a less noisy explanation.

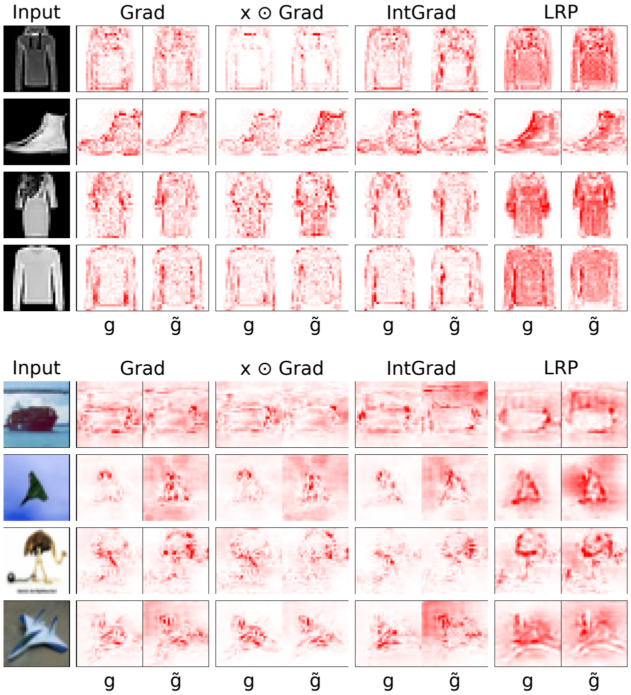

Figure 5. Tsp-explanations for the models and images of Figure 2. The tsp-explanations of the original model g and manipulated $\tilde{g}$ are similar suggesting that the manipulations were mainly due to components orthogonal to the data manifold.

If the adversaries knew that tsp-explanations are used, they could also try to train a model $\tilde{g}$ which manipulates the tsp-explanations directly. However, tsp-explanations are considerable more robust to such manipulations, as shown on the right-hand-side of Figure 3.

We refer to Appendix D for more detailed discussion.

4. Conclusion

A central message of this work is that widely-used explanation methods should not be used as proof for a fair and sensible algorithmic decision-making process. This is because they can be easily manipulated as we have demonstrated both theoretically and experimentally. We propose modifications to existing explanation methods which make them more robust with respect to such manipulations. This is achieved by projecting explanations on the tangent space of the data manifold. This is exciting because it connects explainability to the field of manifold learning. For applying these methods, it is however necessary to estimate the tangent space of the data manifold. For high-dimensional datasets, such as ImageNet, this is an expensive and challenging task. Future work will try to overcome this hurdle. Another promising direction for further research is to apply the methods developed in this work to other application domains such as natural language processing.

Acknowledgements

We thank the reviewers for their valuable feedback. P.K. is greatly indebted to his mother-in-law as she took care of his sick son and wife during the final week before submission. We acknowledge Shinichi Nakajima for stimulating discussion. K-R.M. was supported in part by the German Ministry for Education and Research (BMBF) under Grants 01IS14013A-E, 01GQ1115, 01GQ0850, 01IS18025A and 01IS18037A. This work is also supported by the Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (No. 2017-0-001779), as well as by the Research Training Group ”Differential Equation- and Data-driven Models in Life Sciences and Fluid Dynamics (DAEDALUS)” (GRK 2433) and Grant Math+, EXC 2046/1, Project ID 390685689 both funded by the German Research Foundation (DFG).

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I. J., Hardt, M., and Kim, B. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pp. 9525–9536, 2018.

Aïvodji, U., Arai, H., Fortineau, O., Gambs, S., Hara, S., and Tapp, A. Fairwashing: the risk of rationalization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 161–170. PMLR, 2019. URL http://proceedings.mlr.press/v97/aivodji19a.html.

Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., Samek, W., Müller, K.-R., Dähne, S., and Kindermans, P. iNNvestigate neural networks! *Journal of Machine Learning Research* ***20***, 2019.

Ancona, M., Ceolini, E., Oztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *6th International Conference on Learning Representations (ICLR 2018)*, 2018.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7):1–46, 07 2015. doi: 10.1371/journal.pone.0130140. URL https://doi.org/10.1371/journal.pone.0130140.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

Dombrowski, A.-K., Alber, M., Anders, C., Ackermann, M., Müller, K.-R., and Kessel, P. Explanations can be manipulated and geometry is to blame. In *Advances in Neural Information Processing Systems*, pp. 13567–13578, 2019.

Ghorbani, A., Abid, A., and Zou, J. Y. Interpretation of neural networks is fragile. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.*, pp. 3681–3688, 2019.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Heo, J., Joo, S., and Moon, T. Fooling neural network interpretations via adversarial model manipulation. In *Advances in Neural Information Processing Systems*, pp. 2921–2932, 2019.

Kindermans, P., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. The (un)reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 267–280. Springer, 2019.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Reynolds, J., Melnikov, A., Lunova, N., and Reblitz-Richardson, O. Pytorch captum. https://github.com/pytorch/captum, 2019.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10:1096, 2019.

Lee, J. M. *Introduction to Smooth Manifolds*. Springer, 2012.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., and Müller, K.-R. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019. ISBN 978-3-030-28953-9. doi: 10.1007/978-3-030-28954-6.

Shao, H., Kumar, A., and Thomas Fletcher, P. The riemannian geometry of deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 315–323, 2018.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 3145–3153, 2017. URL http://proceedings.mlr.press/v70/shrikumar17a.html.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6034.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 3319–3328, 2017. URL http://proceedings.mlr.press/v70/sundararajan17a.html.