A. Proofs

A.1. Theorem 1

We first recall a few basic definitions and theorems necessary for the proof of Theorem 1. Our presentation will be necessarily brief as it can hardly replace a course on differential geometry. However we closely follow (Lee, 2012) to which we refer for a more detailed and complete presentation.

Definition 2 *An embedded submanifold S of M is a subset $S \subset M$ that is itself a manifold (with respect to the subspace topology) endowed with a smooth structure with respect to which the inclusion map $S \hookrightarrow M$ is a smooth embedding. If S is closed as a set, the submanifold is called properly embedded.*

Let U be an open subset of $\mathbb{R}^n$ and $k \in \{1, \dots, n\}$. A *k-slice* of U is any subset $S \subset U$ of the form

$$S = \{(x^1, \dots, x^k, 0, \dots 0) \in U\}\,.$$

We say that a submanifold $S \subset M$ satisfies the local k-slice condition if each point $p \in S$ is contained in the domain of a chart (U, ϕ) for which $\phi(S \cap U)$ is a single k-slice in $\phi(U)$.

Theorem 3 *An embedded k-dimensional submanifold S satisfies the local k-slice condition.*

We refer to Theorem 5.8 of (Lee, 2012) for a proof.

Definition 3 *Let M be a smooth manifold and $S \subset M$ an embedded submanifold. A vector field X along S assigns to each $p \in S$ a vector $X_p \in T_pM$.*

For each $p \in S$, we can decompose the tangent space $T_pM = T_pS \oplus T_pS^\perp$, where $T_pS^\perp$ is the orthogonal complement of T_pS.

A standard tool for extending functions from a local coordinate patch to the entire manifold is given by the following definition:

Definition 4 *Let M be a topological space and $\Phi = (\phi_\alpha)_{\alpha \in I}$ an open cover indexed by the set I. A partition of the unity subordinate to Φ is an family $(\psi_\alpha)_{\alpha \in I}$ of continuous functions $\psi_\alpha : M \to \mathbb{R}$ with the properties:*

1. *$\forall x \in M$ and $\forall \alpha \in I$: $0 \leq \psi_\alpha(x) \leq 1$*
2. *$\forall \alpha \in I$: supp$(\psi_\alpha) \subset \phi_\alpha$*
3. *(supp $\psi_\alpha)_{\alpha \in I}$ is locally finite, i.e. $\forall p \in M$, $\exists U \subset M$ such that $U \cap$ supp$(\psi_\alpha) \neq \emptyset$ for only finitely many values of α.*

It can be shown that for any open cover of a manifold M, a partition of the unity subordinate to this cover exists. We refer to Theorem 2.23 of (Lee, 2012) for a proof.

Our main theorem is a generalization of the well-known submanifold extension lemma (see, for example, Lemma 5.34 in (Lee, 2012)). While we could not find such a generalization in the literature, we suspect that it is entirely obvious to differential geometers but typically not needed for their purposes. We now state this main theorem before giving a proof:

Theorem 4 *Let $S \subset M$ be a properly embedded d-dimensional submanifold of the D-dimensional manifold M and $V = \sum_{i=d+1}^D v^i \partial_i$ a smooth vector field along S which for each $p \in S$ assigns vectors in $T_pM^\perp$. For any smooth function $f : S \to \mathbb{R}$, there exists a smooth extension $F : M \to \mathbb{R}$ such that $F|_S = f$ and*

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

for $x \in S$.

Proof: Since S is embedded, there exists a slice chart (U_p, ϕ_p) for each $p \in S$. We extend f in U_p by the smooth map

$$F_p(x_1, \dots, x_D) = f(x_1, \dots, x_d) + \sum_{I=d+1}^{D} v^I(x_1, \dots, x_d)\, x^I\,.$$

By the definition of a slice chart, $\phi(p) = (x_1, \dots, x_d, 0, \dots, 0)$ for $p \in S$. Therefore, it follows that

$$F|_S = f\,.$$

Let $\{\psi_p, p \in S\} \cup \{\chi\}$ be a partition of unity subordinate to the open cover $\{U_p;\, p \in S\} \cup \{M \setminus S\}$.[8] We define

$$F(x) = \sum_{p \in S} \psi_p(x) F_p(x)\,.$$

For $x \in S$, it holds that $F_p(x) = f(x)$ and thus $F(x) = f(x) \sum_{p\in S} \psi_p(x) = f(x)$ because $\sum_{p \in S} \psi_p(x) = 1$. Since the collection of supports of the ψ_p is locally finite, F is smooth.

The gradient of F at $x \in S$ can be straightforwardly calculated. For $I \in \{d+1, \dots, D\}$, one obtains

$$\begin{aligned}\nabla_I F(x) &= \nabla_I \sum_p \psi_p(x) F_p(x) \\ &= \sum_p \nabla_I \psi_p(x)\, F_p(x) + \sum_p \psi_p(x) \nabla_I F_p(x) \\ &= f(x)\, \nabla_I \sum_p \psi_p(x) + \sum_p \psi_p(x) v^I(x)\,.\end{aligned}$$

[8] We note that $M \setminus S$ is open since S is closed.

We note that sum and differentiation commute due to the local finiteness of the partition ψ. Using $\sum_p \psi_p(x) = 1$, it follows that $\partial_I \sum_p \psi_p(x) = 0$. We thus have derived that

$$\partial_I F(x) = v^I(x) \sum_p \psi_p(x) = v^I(x)\,.$$

For $i \in \{1, \dots, d\}$, one obtains

$$\begin{aligned}
&\nabla_i \sum_p \psi_p(x) F_p(x) \\
=&\sum_p \nabla_i \psi_p(x)\, F_p(x) + \sum_p \psi_p(x)\, \nabla_i F_p(x) \\
=&f(x)\, \nabla_i \sum_p \psi_p(x) + \\
&\quad + \sum_p \psi_p(x) \left(\nabla_i f(x) + \sum_{I=d+1}^{D} x^I\, \nabla_i v^I(x) \right) .
\end{aligned}$$

The first term vanishes due to $\nabla_i \sum_p \psi_p(x) = \nabla_i 1 = 0$. For the last term, we use that for $x \in S$ it holds that $x^I = 0$. As a result, we derive that

$$\nabla_i F(x) = \nabla_i f(x)\,.$$

□

A.2. Theorem 2

A.2.1. Bounds on Explanations

As noted in the main text, a global rescaling of the explanation maps h is merely conventional. A natural convention is to bound the explanations such that $h_i \in [-0.5, 0.5]$ for all $i = 1 \dots D$. For the gradient map, this can be ensure by defining $h(x) = \lambda \, \nabla g(x)$ where $\lambda = \frac{1}{C}$ (since by assumption $|\nabla_i g(x)| \leq C$). In particular, all target explanation maps are then chosen to obey this bound. For convenience, we can absorb rescaling λ in the classifier g by redefining $g \to \lambda g$. As a result, we always choose the convention that $|\nabla_i g(x)| \leq 1$ without loss of generality.

More generally, let h denote any bounded explanation method

$$|h_i(x)| \leq C \in \mathbb{R}_+ \qquad \forall x \in S \tag{20}$$

We note that all considered explanation maps obey

$$g \to \lambda g \qquad \Rightarrow \qquad h_g \to \lambda h_g \tag{21}$$

for $\lambda \in \mathbb{R}$ since they are linear in g.

From this, it follows that any bounded explanation method can be assumed to be bounded by 0.5 because this can be ensured by an irrelevant rescaling. We again adopt the convention in which this rescaling factor is absorbed in g.

A.2.2. Proofs for other Explanations

In this appendix, we will proof Theorem 2 for $x \odot$ Grad and ϵ-LRP.

x $\odot$ Grad: We assume that the explanation map of g is bounded, i.e. $|h^g_i(x)| = |(x \odot \nabla_i g(x))_i| \leq C \in \mathbb{R}_+$ for all $x \in S$. We furthermore assume that there exists a chart for which the coordinates $x_i \neq 0$ are non-vanishing for $i > d$. In practice, this can be easily ensured by an appropriate shift of the data.[9] Given a target explanation $h^t(x)$, we choose a extension G of $g|_S$ such that

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

The explanation of G is given by $h_G(x) = x \odot \nabla G(x)$. The mean-squared error between target and model explanation is then given by

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

This sum can be decomposed as

$$\tfrac{1}{D} \sum_{i=1}^{d} (x_i \nabla_i g(x) - h^t)^2 + \tfrac{1}{D} \sum_{i=d+1}^{D} x_i^2 (\nabla_i G(x) - \tfrac{h^t_i(x)}{x_i})^2$$

Using the fact that we can assume $|h^g_i| = |x_i \nabla_i g(x)| \leq 0.5$ without loss of generality[10] and that we can rescale h^t arbitrarily, it then follows

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

ϵ-LRP: We assume that the network uses relu non-linearities. In fact, LRP can be shown to be theoretically well-motivated under this assumption by using Deep Taylor Decomposition (Montavon et al., 2017).

It can be shown that ϵ-LRP can be mathematically reformulated as

$$h_{\epsilon\text{LRP}} = x \odot \tilde{\nabla} g(x)\,,$$

where the operator $\tilde{\nabla}$ acts on non-linearities f by

$$\tilde{\nabla} f(z) = \tfrac{f(z)}{z} \tag{22}$$

and on affine linear functions as the standard gradient ∇. We refer to the Appendix A of (Ancona et al., 2018) for a

[9] If we do not allow for the freedom of shifting the data, any valid $\mathbf{x} \odot$ Grad explanation map must have zero relevance for input components x_i which are vanishing. If one restrict the target map h^t to be valid, no shifts are needed for the proof.

[10] We note that the necessary rescaling of h^g is not in conflict with the shift to ensure $x_i \neq 0$ because the latter condition is scale-invariant.

proof. By our assumption, all non-linearities are relu and therefore obey

$$\tilde{\nabla}\text{relu}(x) = \theta(x)$$

where $\theta(x)$ is the Heaviside step function. This coincides with normal gradient operator $\nabla\text{relu}(x) = \theta(x)$. This observation was, to the best of our knowledge, first made in (Ancona et al., 2018). Therefore, the proof for $x \odot$ Grad applies verbatim for this method as well. □

A.3. Flat Manifolds and other Explanation Methods

It was shown in the main text that one can always construct a model

$$\tilde{g}(x) = \sigma\left(w^T x + \sum_i \lambda_i (\hat{w}^{(i)^T} x - b_i) + c\right) , \qquad (23)$$

which agrees with $g(x) = \sigma(w^T x + c)$ for all datapoints $x \in S$ but has gradient explanation map

$$h_{\text{grad}}(x) = w + \sum_i \lambda_i \hat{w}^{(i)} . \qquad (24)$$

By choosing λ_i appropriately, we can always set components of h_{grad} corresponding to orthogonal directions $\hat{w}_i$ of the data S to an arbitrary h^t_i, i.e.

$$\lambda_i = h^t_i - w^T \hat{w}^{(i)}$$

where we have normalized $\hat{w}^{(i)}$ such that it has unit norm. For $x \odot$ Grad, we can similarly choose

$$\lambda_i = \frac{h^t_i - (x \odot w)^T \hat{w}^{(i)}}{(x \odot \hat{w}^{(i)})^T \hat{w}^{(i)}}$$

As already discussed in Appendix A.2, valid $x \odot$ Grad explanations map have to be zero in components h_i for which the corresponding input component x_i are vanishing. As a result, one only needs to set λ_i to a non-vanishing value if $x_i \neq 0$. Thus, the expression above is well-defined for all valid explanation maps. The corresponding statement for ϵ-LRP method can be proven completely analogously.

We also note that ϵ-LRP and IntGrad coincide with the x$\odot$Grad method for logistic regression. For the latter, one has to choose a vanishing baseline point $\bar{x}$. The generalization to non-vanishing baselines is however straightforward by substituting $x \to x - \bar{x}$.

B. Credit Risk using other Explanation Methods

We originally tested our procedure on two credit-risk datasets. Unfortunately, we realized that the licences of these datasets do not permit publication of these results. Since our results only mildly depend on the data (for example, the gradient explanation is completely independent of it), we decided to generate a synthetic dataset as follows: the feature 'gender' is sampled with equal probability for the values 1 for male or -1 for female. The feature 'income' is sampled from a normal distribution with mean $\mu = 5000$ and standard deviation $\sigma = 5000$. We clipped to a minimum of 250 to ensure only positive income. We then normalized the income to take values between 0 and 1 by dividing by the maximum income. The feature 'taxes' is $0.4 x_{\text{income}}$ and, for simplicity, not further normalized. We use $\lambda = 1000$ as scaling factor for the weights $\hat{w}$ of the modified classifier $\tilde{g}$.

The bars in Figures 6 and 8 show the average explanation map with error bars as standard deviations. We only show explanation maps for positive classification results (examples where credit was given). All explanation maps are normalized to have $\sum_i |h_i| = 1$.

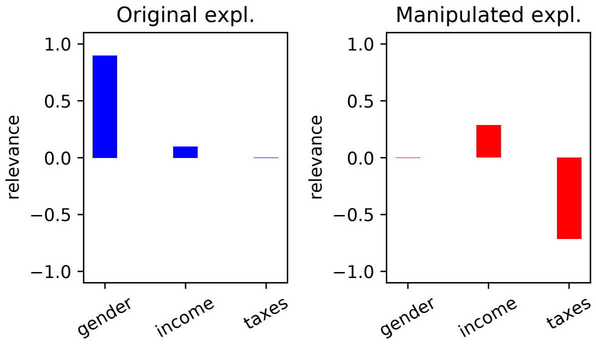

Figure 6. Gradient explanations for classifier g and fairwashed classifier $\tilde{g}$ highlight completely different features.

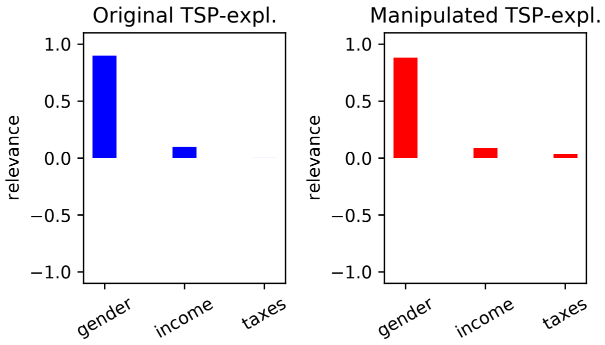

Figure 7. Gradient *tsp-explanations* for **original classifier g** and **manipulated $\tilde{g}$** highlight the same features.Colored bars show the median of the explanations over multiple examples.

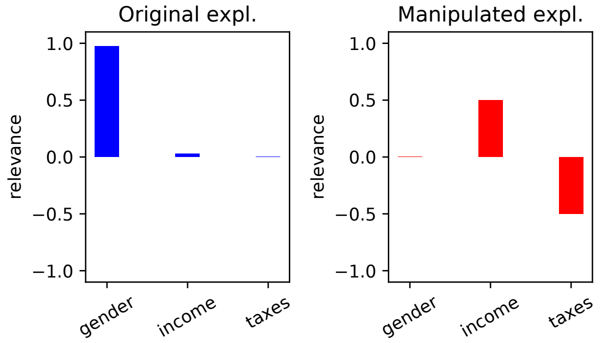

Figure 8. x⊙Grad explanations for classifier g and fairwashed classifier $\tilde{g}$ highlight completely different features.

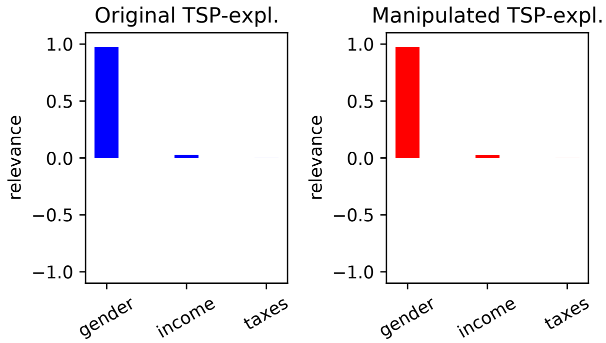

Figure 9. x⊙Grad *tsp-explanations* for **original classifier g** and **manipulated $\tilde{g}$** highlight the same features. Colored bars show the median of the explanations over multiple examples.

C. TSP-Explanations

For the $x \odot$ Grad method, we let the projection operator act only on the gradient factor of the explanation map, i.e.

$$\hat{h}_{\text{xGrad}}(x) = x \odot P\,\nabla g(x)\,. \tag{25}$$

This is equivalent to redefining the projection matrix to

$$P_{ij} \to \begin{cases} \frac{x_i}{x_j} P_{ij} & \text{for } x_j \neq 0\,, \\ 0 & \text{for } x_j = 0\,. \end{cases} \tag{26}$$

and applying this redefined projection operator on the unprojected map h_{xGrad}, i.e.

$$\hat{h}_{\text{xGrad}}(x) = P\,h_{\text{xGrad}}(x)\,. \tag{27}$$

Analogously, we define for the IntGrad method

$$\hat{h}_{\text{IntGrad}}(x) = (x - \bar{x}) \odot \frac{1}{N} \sum_{k=0}^{N} P\,\nabla g\left(\bar{x} + \tfrac{k}{N}(x - \bar{x})\right)\,, \tag{28}$$

where P projects on the tangent space of the point at which the corresponding gradient is calculated. In practice however, we cannot guarantee that all the corresponding points lie on the data manifold S. We therefore propose to use the projection operator for the data point x instead. We find empirically that this leads to robuster explanations. This definition can again be reformulated in terms of a redefinition of the projection operator in complete analogy to the case of $x \odot$ Grad.

For the LRP method, we propose to use the generalized projection matrix (26) since ϵ-LRP is equivalent to $x \odot$Grad for relu activations (see Appendix A.2) but we also find empirically that the standard projection matrix on the data manifold leads to more robust explanations.

C.1. Flat manifold and Logistic Regression

For $x \odot$Grad method, we again straightforwardly see that the tsp-explanations for g and $\tilde{g}$ agree by applying the definition (25), i.e.

$$\hat{h}_g(x) = x \odot P\,\nabla g(x) = x \odot P\,\nabla \tilde{g}(x) = \hat{h}_{\tilde{g}}(x)\,. \tag{29}$$

The corresponding statement for ϵ-LRP can be proven analogously. The same is true for IntGrad if one assumes that all intermediate point as well as the baseline point are on the data manifold.

C.2. Autoencoder Method

In the following, we will first show how the proposed procedure for estimating tangent space arises from certain asymptotic limit of autoencoders.

Definition 5 *An asymptotically-trained autoencoder with encoder $E : M \to Z$ and $D : Z \to M$ has zero reconstruction error, i.e.*

$$E_{rc} = \int_S d^D x\, p_{data}(x)\, ||(D \circ E)(x) - x||^2 = 0\,,$$

where p_{data} is a continuous probability density describing the data. Furthermore, the decoder maps on the data manifold S, i.e.

$$\forall z \in Z : \qquad D(z) \in S\,.$$

The latter condition arises from the fact that we want the decoder to generate data samples from latent representations. We note there is good theoretical and experimental evidence that these conditions hold asymptotically for (at least some of the) popular autoencoder architectures, in particular Variational Autoencoders (Kingma & Welling, 2014).

Theorem 5 *For a continuous data distribution p_{data}, it holds that*

$$E_{rc} = 0 \quad \Rightarrow \quad \forall x \in S: \; x = (D \circ E)(x)\,, \qquad (30)$$

i.e. every datapoint x is perfectly reconstructed.

Proof: Suppose, there exists a $x_0 \in S$ such that $x_0 \neq (D \circ E)(x_0)$. Since the integrand of E_{rc} is continuous, we can always find an $\epsilon > 0$ such that this condition holds for every $x \in [x-\epsilon, x+\epsilon]$. Let $\Delta \in \mathbb{R}_+$ denote the infimum of the integrand on this interval. By positivity of the integrand, it holds that $E_{\text{rc}} \geq 2\epsilon\,\Delta > 0$. □

This theorem then immediately implies that:

Theorem 6 *The decoder $D : Z \to S$ of an asymptotically-trained autoencoder is surjective on the data manifold S.*

Proof: Assume the contrary, then there exists a $x \in S$ such that $\nexists z \in Z$: $D(z) = x$. But by the previous theorem, it has to hold that $z = E(x)$ obeys $D(z) = x$ since the autoencoder has vanishing reconstruction error.□

The differential $\mathrm{d}_z D(z) = \frac{\partial D}{\partial z}(z)$ is a linear map from the tangent space of Z to the tangent space of S, i.e. $\mathrm{d}_z D(z) : T_z Z \to T_{D(z)} S$. Since the decoder is surjective, the rank of $\mathrm{d}_z D$ is the same as the dimensionality of the data manifold S, i.e. $\text{rk}(d_z D) = d$. These are basic facts of differential geometry and we refer to Chapter 5 and 6 of (Lee, 2012) for a detailed discussion. As a result, the left-singular vectors $u_1, \dots u_d \in \mathbb{R}^D$, corresponding to the d non-vanishing singular values of the decomposition $d_z D(z) = U \Sigma V$, span the data tangent space $T_{D(z)} S$.

In the non-asymptotic limit, it cannot be expected that this relation holds exactly. For a sufficiently well-trained autoencoder, it is however reasonable to expect that the left-singular values $u_1, \dots, u_d \in \mathbb{R}^D$ corresponding to the d largest singular values are a good approximation for the basis of the data tangent space.

We stress however that we do not have a rigorous proof for this outside of the asymptotic regime discussed above. We furthermore want to remark that our thinking was heavily inspired by the discussion in (Shao et al., 2018) which uses very similar techniques. Last but not least, there are a number of alternative approaches in the literature to estimate tangent space. Notable examples include Contractive Autoencoders (Rifai et al., 2011) and semi-supervised GANs (Kumar et al., 2017). It would be interesting to compare these approaches to the one taken in this paper but we leave this to future work.

D. Details on Experiments

Model Architecture: For FashionMNIST and MNIST, we used a convolutional network with two groups of convolution with 20 and 50 filters of size 5×5 respectively, relu activation and max-pooling over 2×2, followed by a dense layer with 500 outputs, a relu activation, and finally another dense layer with outputs down to the number of classes (10). We used VGG16 (Simonyan & Zisserman, 2015) for experiments on CIFAR10.

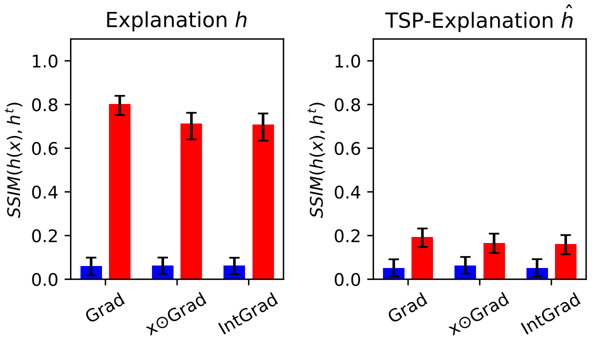

Figure 10. **Left:** SSIM of the target map h^t and explanations of **original model** g and **manipulated** $\tilde{g}$ respectively. **Right:** Same as on the left but for *tsp-explanations*. The model $\tilde{g}$ was trained to manipulate the tsp-explanation, but this time with a higher weighting factor $\gamma = 9$. Even with this more aggressive manipulation compared to the original experiment in Figure 3, 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.

Model Training: All images were normalized to mean 0 and standard deviation 1 within the training set over all pixels. For CIFAR10 training, we padded all images with 4 pixels of each side in every dimension, and then randomly cropped back to the original size of 32×32.

The original models for FashionMNIST and MNIST were trained from scratch using standard SGD with a learning rate of 0.01 and a momentum of 0.5. The original VGG-16 model for CIFAR10 was trained also trained using standard SGD, but with a learning rate of 0.05, momentum of 0.9 and weight decay of 5×10^{-4}.

All manipulated models on all datasets were trained using Adam (Kingma & Ba, 2015) by fine-tuning the original model with a fixed learning rate of 10^{-5} until convergence. We set the weighting factor γ of the loss function (11) to 4. We use the same hyperparameters for manipulating tsp-explanations to ensure fair comparison. To ensure our results do not depend on a specific weighting factor γ, we demonstrate the same experiment shown in Figure 3 with $\gamma = 9$ in Figure 10.

Target Explanation: The target explanation map used in our experiments is shown in Figure 11.

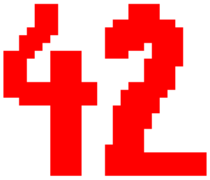

Figure 11. Image used as the target explanation to train the manipulated models.

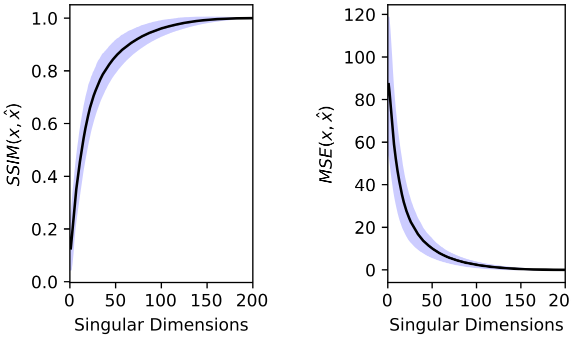

Figure 12. SSIM (left) and MSE (right) of original vs. reconstructed image from the tangent-space directions, ordered by number of used directions. Images are drawn from the full FashionMNIST test set. A total of 200 neighbours was used for each image. The black curve describes the median, while the surrounding blue area marks the space between the 25th and 75th percentiles.

Model Statistics: The accuracies, MSE and KL-divergence of the original and adversarially trained models are documented in Tables 1, 2 and 3 respectively.

Estimating Tangent Space: In the following, we briefly summarize the procedure used to estimate tangent space for the various datasets.

MNIST, FMNIST: We use the hyperplane method described in the main text. For a given data point, the nearest neighbours are taken only from the training set. The dimensionality of the hyperplane is chosen to be 30. This number was tuned by ensuring that the data points are well reconstructed with respect to the MSE (which corresponds to the Euclidean distances, i.e. the natural metric on the embedding space $\mathbb{R}^D$), see Figure 12. The hyperplane is fitted using the nearest neighbours and the datapoint itself. Before fitting, all datapoints are normalized to have zero mean and a standard deviation of one.

CIFAR10: We use the autoencoder method described in the main text. This is because the manifold is not densly sampled enough for the hyperplane method, see Figure 13. We normalize the data as described above and split it by class.

Method	MNIST	FashionMNIST	CIFAR10
Original	98.97	94.72	92.47
Gradient	98.84	94.58	91.77
x $\odot$ Grad	98.96	94.48	91.53
IntGrad	98.95	94.65	91.62
LRP	98.95	94.68	92.08

Table 1. Accuracies of all models in percent.

A separate autoencoder is trained for each class for three epochs using the Adam optimizer with a learning rate of 0.001. We use a same VQ-VAE architecture as in this example[11]. After training, the Jacobian $\frac{\partial D}{\partial z_e}(x)$ is calculated by backpropagation for each data sample x. We note that this could be sped up by forward-mode differentiation. We then perform an SVD-decomposition of the result and tune the number of singular components ensuring good reconstruction.

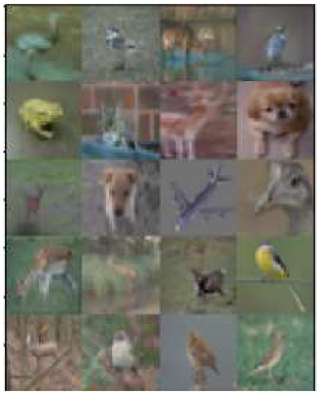

Figure 13. Nearest neighbours with respect to Euclidean distance for image in the top left-hand corner. Clearly, the neighbours are not local deformation of the image itself. As a result, the hyperplane method cannot be used for the CIFAR10 dataset.

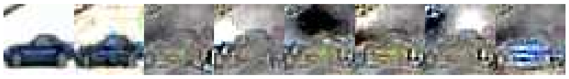

Figure 14. The input image is shown on the very left. Second image is the reconstruction from the tangent-space directions of which six are shown on the right.

D.1. FashionMNIST

D.1.1. Additional Heatmaps

[11] https://github.com/deepmind/sonnet/blob/master/sonnet/examples/vqvae_example.ipynb

Method	MNIST	FashionMNIST	CIFAR10
Gradient	120.54	11.13	838.80
x ⊙ Grad	114.29	15.07	933.38
IntGrad	128.03	13.04	707.52
LRP	119.08	3.76	647.45

Table 2. MSE×10^5 of model outputs $g(x)$ and $\tilde{g}(x)$ after final softmax.

Method	MNIST	FashionMNIST	CIFAR10
Gradient	1.21	1.99	8.39
x ⊙ Grad	1.14	2.06	9.34
IntGrad	1.50	2.00	6.30
LRP	1.19	1.19	6.88

Table 3. Mean KL-Divergence×10^3 between models g and $\tilde{g}$.

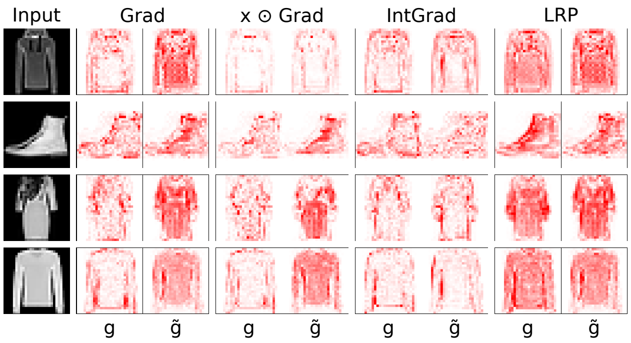

Figure 15. Projected explanations from the original model g (left) and the manipulated model $\tilde{g}$ (right) where the projected heatmaps were attacked for various images from the FashionMNIST test set.

D.1.2. Additional Distance Metrics for Quantitative Comparison

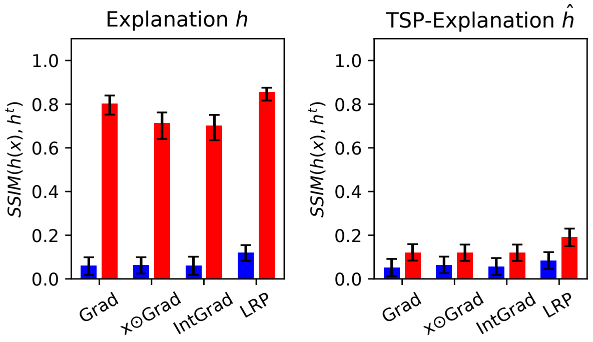

Figure 16. Median of SSIM of left: $h_g(x)$ (blue) and $h_{\tilde{g}}$ (red), right: $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) where $h(x)$ was manipulated, on FashionMNIST.

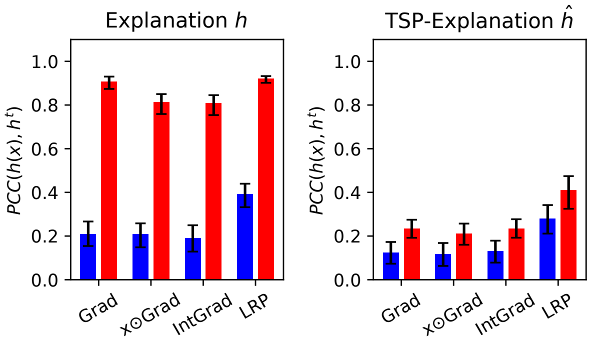

Figure 17. Median of PCC of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on FashionMNIST where $h(x)$ was manipulated.

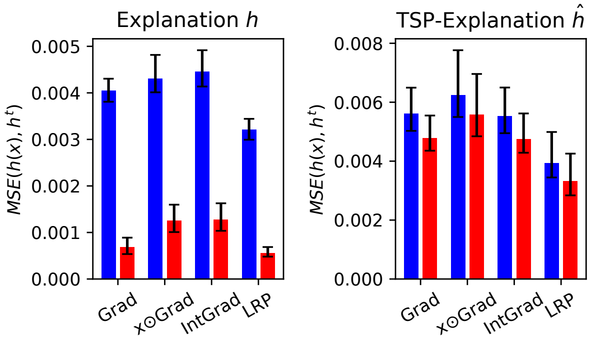

Figure 18. Median of MSE of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on FashionMNIST where $h(x)$ was manipulated.

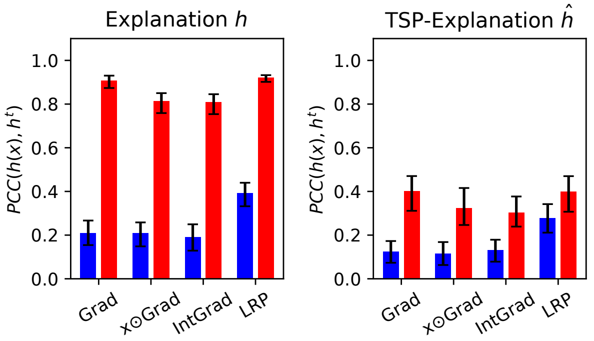

Figure 19. Median of PCC of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on FashionMNIST where $\hat{h}(x)$ was manipulated.

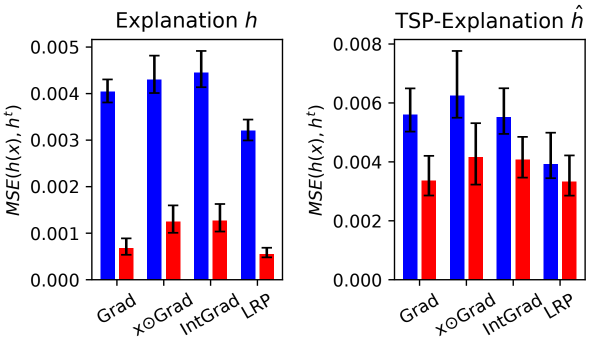

Figure 20. Median of MSE of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on FashionMNIST where $\hat{h}(x)$ was manipulated.

D.2. MNIST

D.2.1. Heatmaps

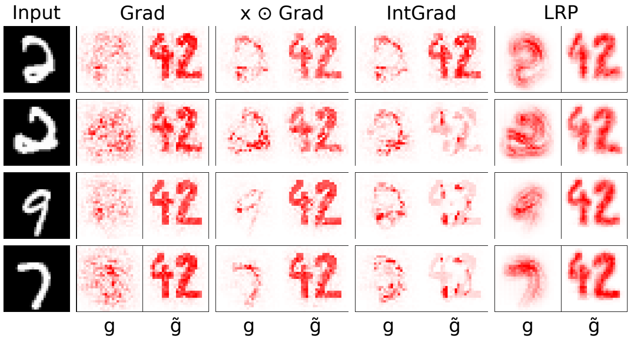

Figure 21. Example explanations from the original model g (left) and the manipulated model $\tilde{g}$ (right) for various images from the MNIST test set.

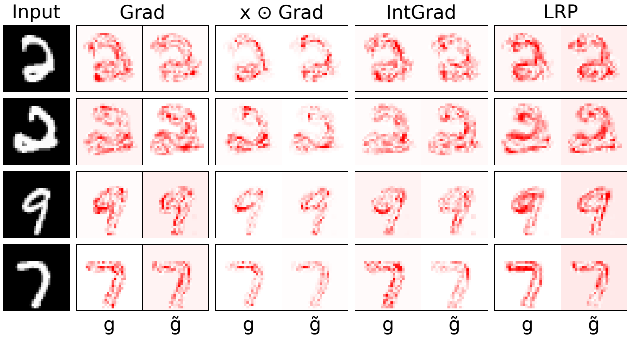

Figure 22. Example tsp-explanations from the original model g (left) and the manipulated model $\tilde{g}$ (right) for various images from the MNIST test set.

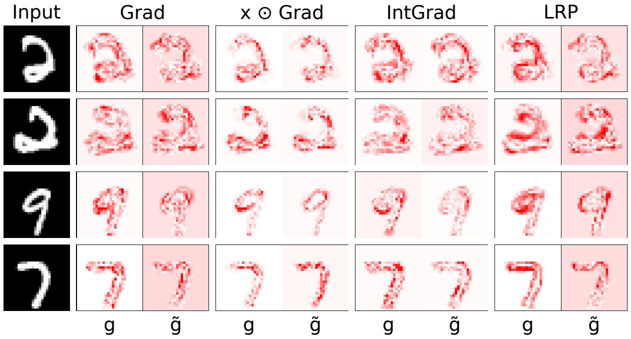

Figure 23. Example tsp-explanations from the original model g (left) and the manipulated model $\tilde{g}$ (right) where the projected heatmaps were attacked for various images from the MNIST test set.

D.2.2. Quantitative Comparison

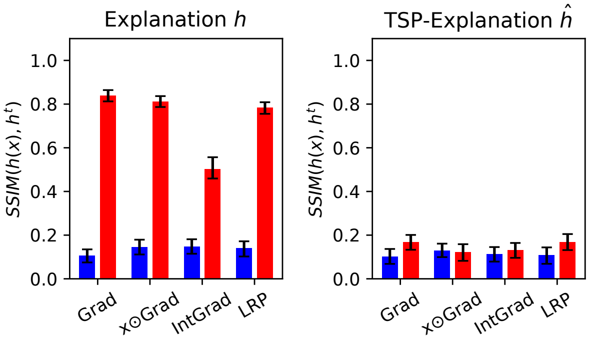

Figure 24. Median of SSIM of left: $h_g(x)$ (blue) and $h_{\tilde{g}}$ (red), right: $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) where $\hat{h}(x)$ was manipulated, on MNIST.

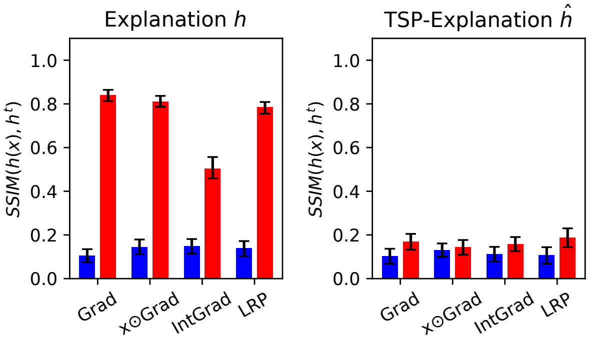

Figure 25. Median of SSIM of left: $h_g(x)$ (blue) and $h_{\tilde{g}}$ (red), right: $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) where $h(x)$ was manipulated, on MNIST.

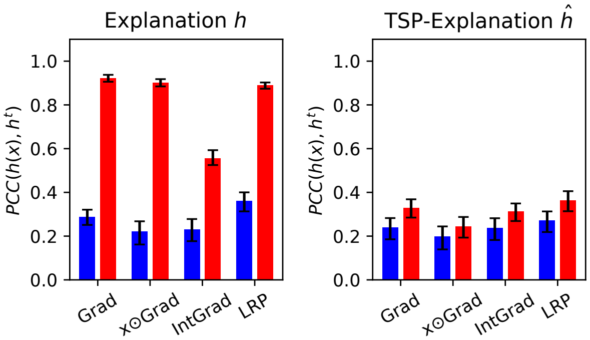

Figure 26. Median of PCC of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on MNIST where $h(x)$ was manipulated.

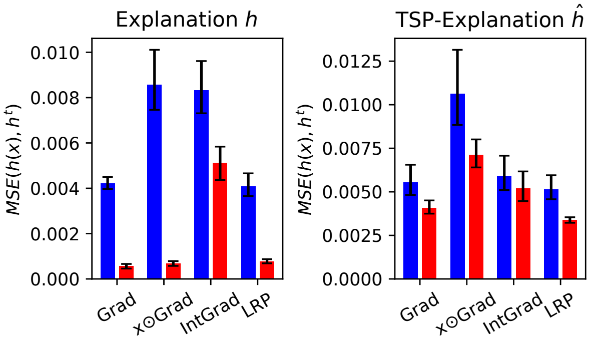

Figure 27. Median of MSE of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on MNIST where $h(x)$ was manipulated.

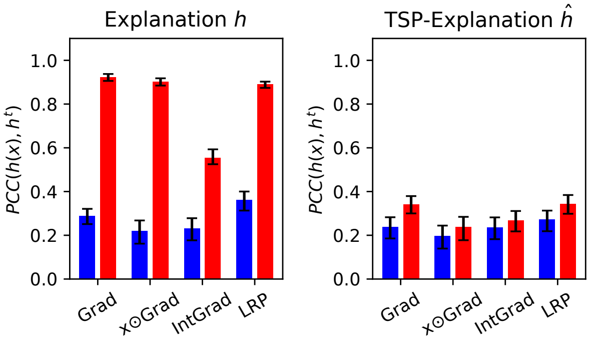

Figure 28. Median of PCC of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on MNIST where $\hat{h}(x)$ was manipulated.

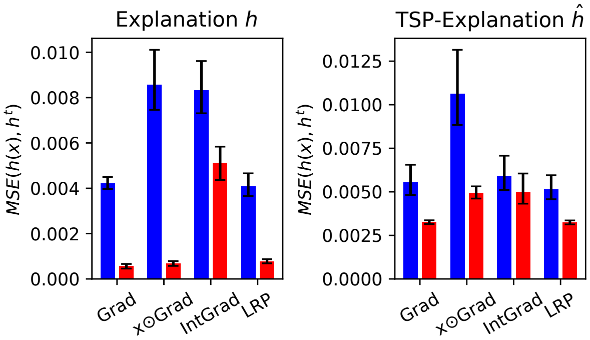

Figure 29. Median of MSE of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on MNIST where $\hat{h}(x)$ was manipulated.

D.3. CIFAR10

D.3.1. Heatmaps

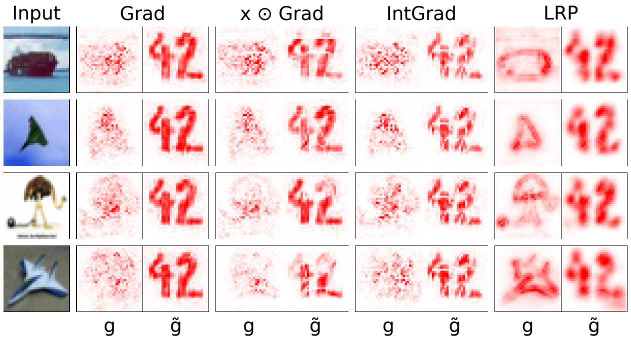

Figure 30. Example explanations from the original model g (left) and the manipulated model $\tilde{g}$ (right) for various images from the CIFAR10 test set.

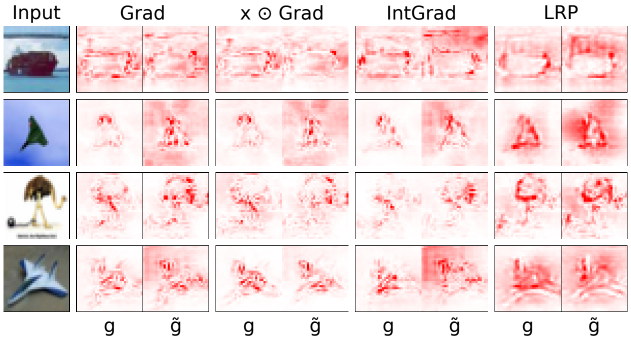

Figure 31. Example tsp-explanations from the original model g (left) and the manipulated model $\tilde{g}$ (right) for various images from the CIFAR10 test set.

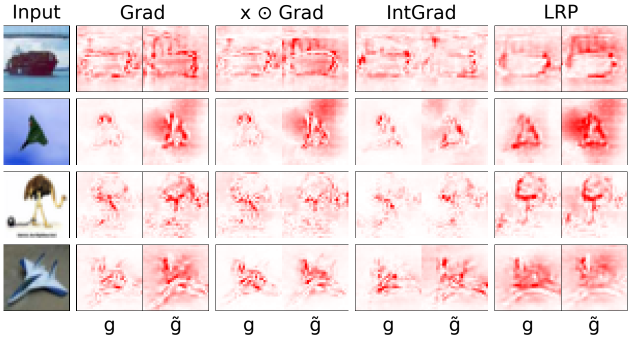

Figure 32. Example tsp-explanations from the original model g (left) and the manipulated model $\tilde{g}$ (right) where the projected heatmaps were attacked for various images from the CIFAR10 test set.

D.3.2. Quantitative Comparison

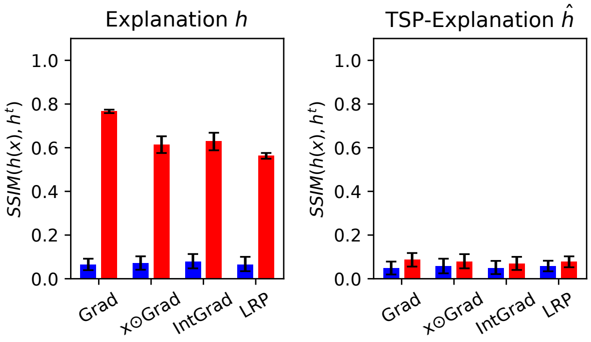

Figure 33. Median of SSIM of left: $h_g(x)$ (blue) and $h_{\tilde{g}}$ (red), right: $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) where $\hat{h}(x)$ was manipulated, on CIFAR10.

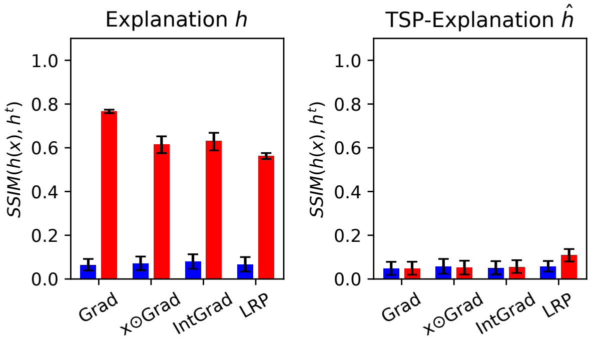

Figure 34. Median of SSIM of left: $h_g(x)$ (blue) and $h_{\tilde{g}}$ (red), right: $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) where $h(x)$ was manipulated, on CIFAR10.

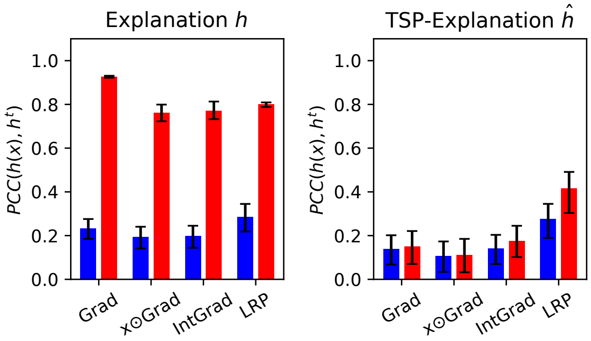

Figure 35. Median of PCC of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on CIFAR10 where $h(x)$ was manipulated.

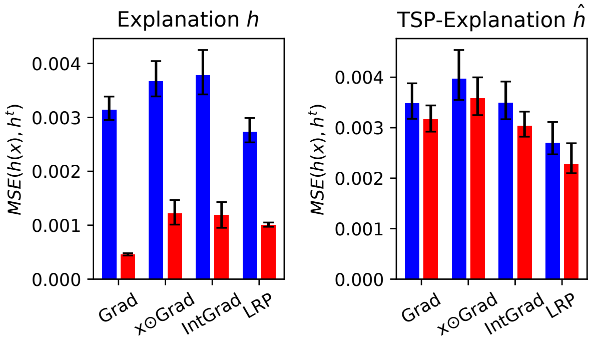

Figure 36. Median of MSE of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on CIFAR10 where $h(x)$ was manipulated.

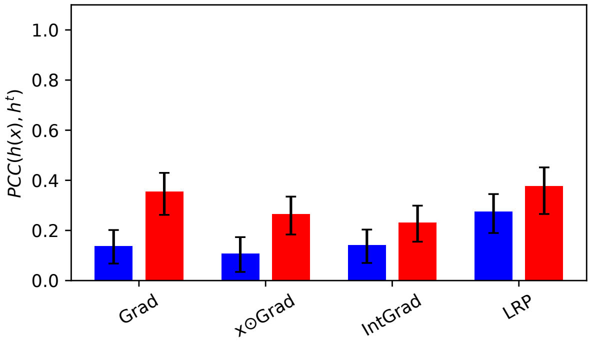

Figure 37. Median of PCC of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on CIFAR10 where $\hat{h}(x)$ was manipulated.

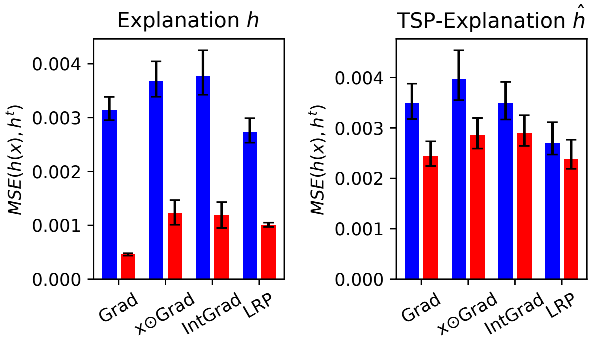

Figure 38. Median of MSE of $\hat{h}_g(x)$ (blue) and $\hat{h}_{\tilde{g}}$ (red) on CIFAR10 where $\hat{h}(x)$ was manipulated.

E. Pixel-flipping

We compare the original explanations with the respective TSP-explanations using *pixel-flipping* (Samek et al., 2017). This metric measures how fast the network confidence $g(x)$ declines when removing features with highest relevance. The pixels are inpainted using the *telea*-method (Telea, 2004) to alleviate uncontrolled behaviour of the classifier off the manifold. Our result clearly show that tsp-methods perform well on this metric.

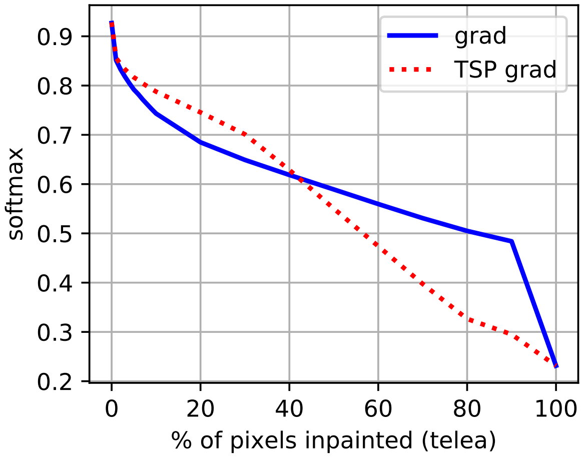

Figure 39. Pixel-flipping performance of Gradient and TSP-Gradient on FashionMNIST

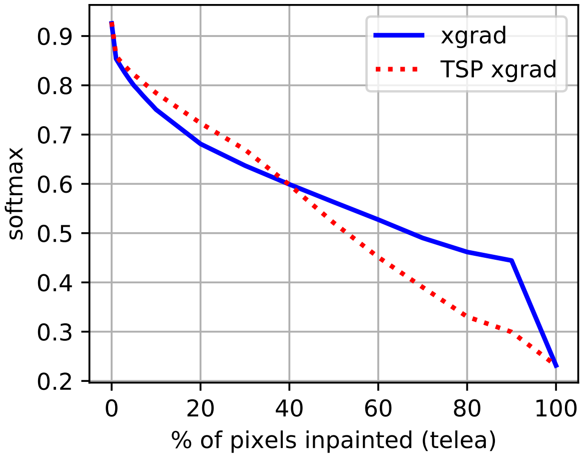

Figure 40. Pixel-flipping performance of x⊙Grad and TSP-x⊙Grad on FashionMNIST

Additional References

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. 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/1412.6980.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April*

14-16, 2014, Conference Track Proceedings, 2014. URL http://arxiv.org/abs/1312.6114.

Kumar, A., Sattigeri, P., and Fletcher, P. T. Improved semi-supervised learning with gans using manifold invariances. *CoRR*, abs/1705.08850, 2017. URL http://arxiv.org/abs/1705.08850.

Rifai, S., Dauphin, Y. N., Vincent, P., Bengio, Y., and Muller, X. The manifold tangent classifier. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pp. 2294–2302, 2011. URL http://papers.nips.cc/paper/4409-the-manifold-tangent-classifier.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28:2660–2673, 11 2017. doi: 10.1109/TNNLS.2016.2599820.

Telea, A. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9, 01 2004. doi: 10.1080/10867651.2004.10487596.