Explanations can be manipulated and geometry is to blame

Authors: Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J. Anders, Marcel Ackermann, Klaus-Robert Müller, Pan Kessel

Presented by: Chelsea (Zixi) Chen, Tessa Han, Vignav Ramesh

Introduction

Motivation

Understand and verify aspects of ML models

Aid decision making in high-stakes scenarios

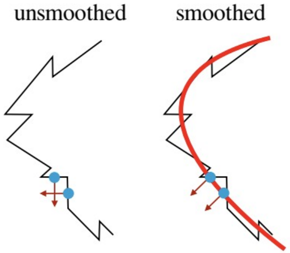
→ Reliable explanations of models!

Can we always trust model explanations?

Summary / Contribution

- Manipulate explanations!
- Provide a theoretical understanding of such nonrobustness and derive a bound
- Introduce smoothing to increase explanation robustness!

$$\|h(p) - h(p_0)\| \leq |\lambda_{max}|\, d_g(p, p_0) \leq \beta\, C\, d_g(p, p_0),$$



Background + Related Work

- Interpretation of Neural Networks is Fragile [Ghorbani et al.]
  - Complex decision boundary
- The (un)reliability of saliency methods [Kindermans et al.]
  - Input invariance
- Sanity checks for saliency maps [Adebayo et al.]
  - Randomization test (Wednesday)

- Fairwashing Explanations with Off-Manifold Detergent [Anders et al.]
  - Low-dimensional data manifold v.s. High-dimensional embedding space

Methodology

Notation

- Neural network $g : \mathbb{R}^d \rightarrow \mathbb{R}^K$ with relu non-linearities
- Classifies input image x into K categories, predicted class $k = \arg\max_i g(x)_i$
- Explanation map: $h : \mathbb{R}^d \rightarrow \mathbb{R}^d$

- Target map: $h^t \in \mathbb{R}^d$
- Manipulated image: $x_{adv} = x + \delta x$

Properties of Manipulated Image

1. The output of the network stays approximately constant, i.e. $g(x_{adv}) \approx g(x)$.
2. The explanation is close to the target map, i.e. $h(x_{adv}) \approx h^t$.
3. The norm of the perturbation δx added to the input image is small, i.e. $\|\delta x\| = \|x_{adv} - x\| \ll 1$ and therefore not perceptible.

Explanation Methods

Gradient-based

- Vanilla gradients: $h(x) = \frac{\partial g}{\partial x}(x)$
  - Quantifies how infinitesimal perturbations in each pixel change the prediction
- Gradient × Input: $h(x) = x \odot \frac{\partial g}{\partial x}(x)$
  - For linear models, this measure gives the exact contribution of each pixel to the prediction
- Integrated Gradients: $h(x) = (x - \bar{x}) \odot \int_0^1 \frac{\partial g(\bar{x} + t(x - \bar{x}))}{\partial x} \mathrm{d}t$

Propagation-based

- Guided Backpropagation
- Layer-wise Relevance Propagation
- Pattern Attribution
  - Standard backpropagation upon element-wise multiplication of the weights with learned patterns

Manipulation Method

- Obtain manipulated images by optimizing the loss function

$$\mathcal{L} = \left\| h(x_{adv}) - h^t \right\|^2 + \gamma \left\| g(x_{adv}) - g(x) \right\|^2$$

manipulated explanation map

target map

weighting hyperparameter

network output (manipulated input)

network output (original input)

with respect to x_{adv} using gradient descent

Manipulation Method

- The gradient with respect to the input ∇ *h(x)* of the explanation often depends on the vanishing second derivative of the relu non-linearities. This causes problems during optimization of the loss function:

$$\partial_{x_{adv}} \left\| h(x_{adv}) - h^t \right\|^2 \propto \frac{\partial h}{\partial x_{adv}} = \frac{\partial^2 g}{\partial x_{adv}^2} \propto \mathrm{relu}'' = 0$$

- **Solution:** replace relu with softplus

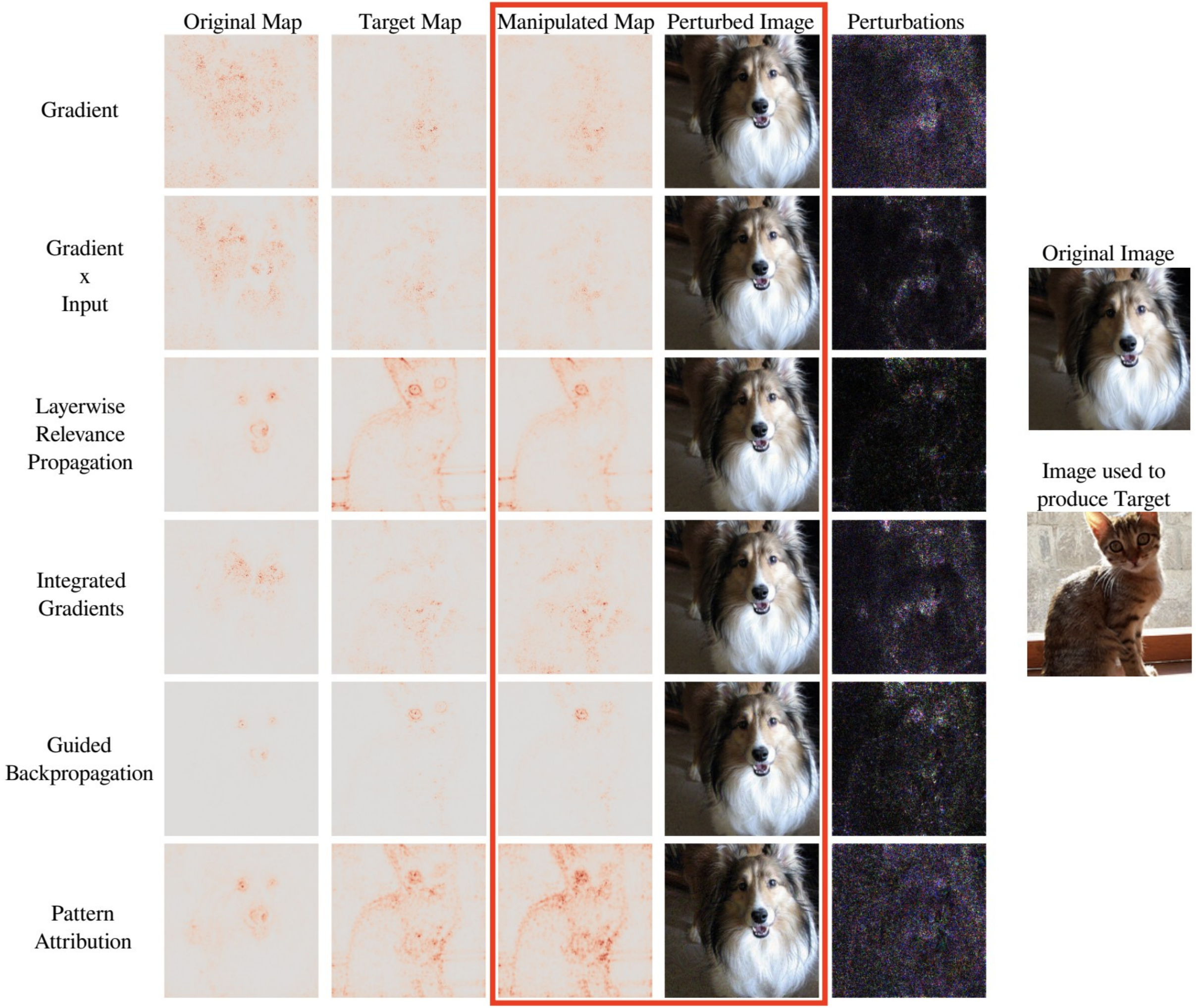
$$\mathrm{softplus}_\beta(x) = \frac{1}{\beta} \log(1 + e^{\beta x})$$

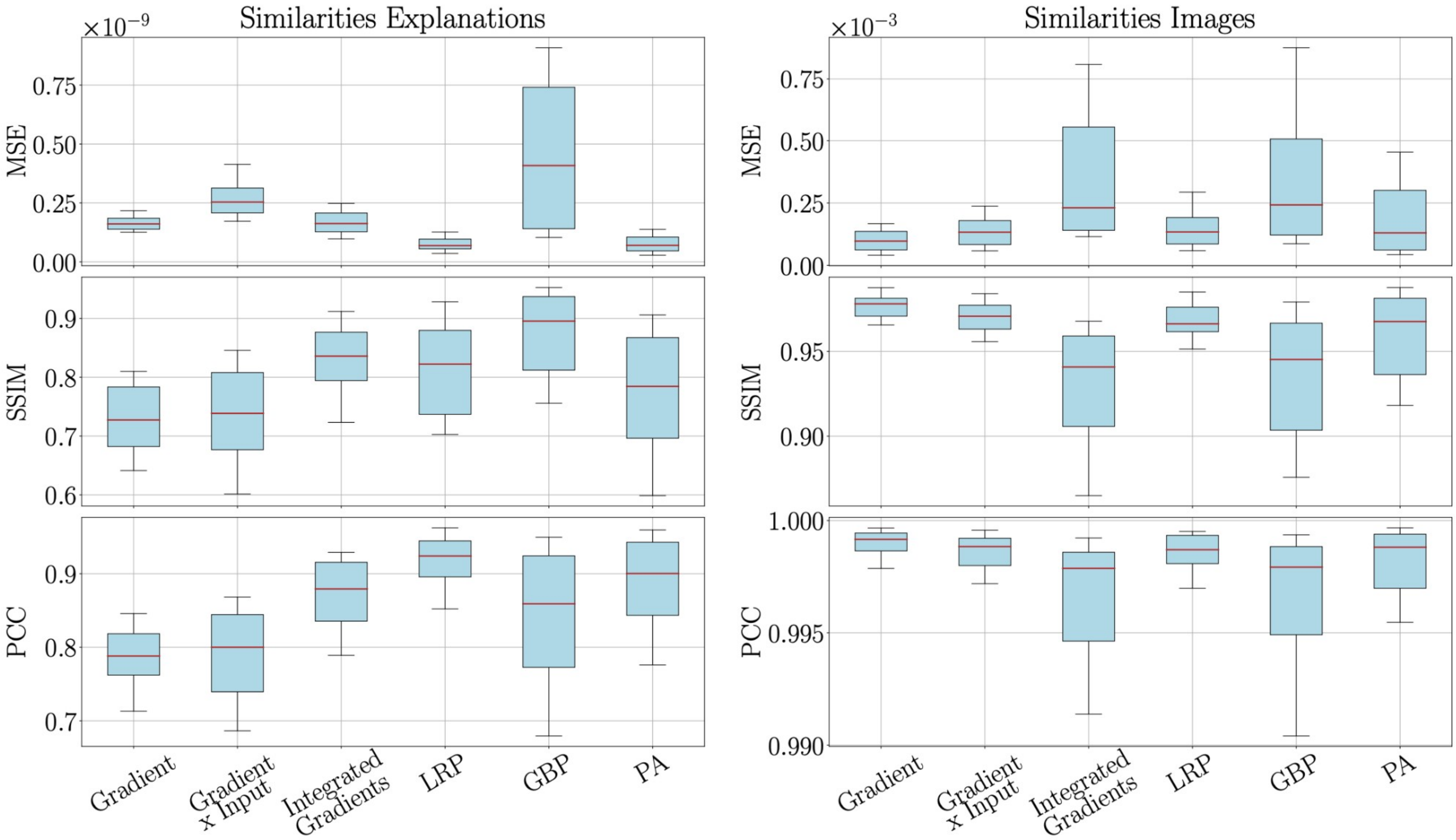
Experiments

Experimental Setup

- Apply algorithm to 100 randomly selected images for each explanation method
- Use VGG-16 network pre-trained on ImageNet
- For each run, we randomly select two images from the test set.
  - One of the two images is used to generate a target explanation map
  - The other image is perturbed by our algorithm with the goal of replicating the target using a few thousand iterations of gradient descent

- Comparable results obtained for ResNet-18, AlexNet, and Densenet-121 + CIFAR-10 dataset


Original Map
Target Map
Manipulated Map
Perturbed Image
Perturbations
Gradient
Gradient x Input
Layerwise Relevance Propagation
Integrated Gradients
Guided Backpropagation
Pattern Attribution
Original Image
Image used to produce Target



Similarities Explanations
×10^-9
MSE
SSIM
PCC
Similarities Images
×10^-3
Gradient
Gradient x Input
Integrated Gradients
LRP
GBP
PA

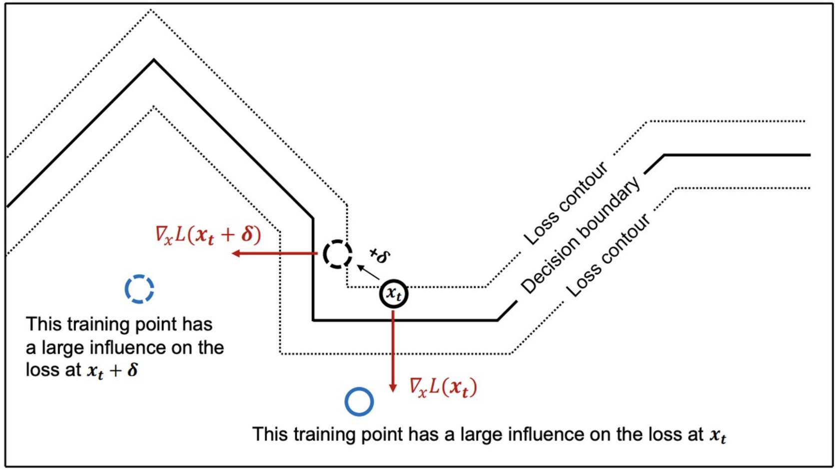
Theoretical Analysis

Intuition

Why are explanations vulnerable and unreliable?

Large curvature of the NN output manifold!

[Ghorbani et al.]



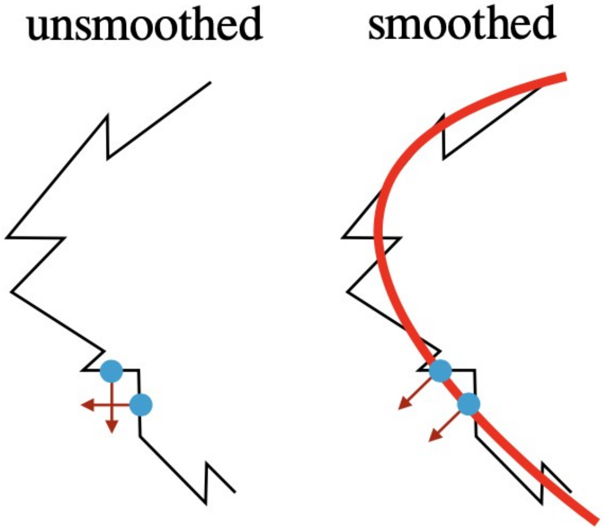
Theoretical Bound

Theorem 1 *Let $g : \mathbb{R}^d \to \mathbb{R}$ be a network with softplus$_\beta$ non-linearities and $\mathcal{U}_\epsilon(p) = \{x \in \mathbb{R}^d; \|x - p\| < \epsilon\}$ an environment of a point $p \in S$ such that $\mathcal{U}_\epsilon(p) \cap S$ is fully connected. Let g have bounded derivatives $\|\nabla g(x)\| \geq c$ for all $x \in \mathcal{U}_\epsilon(p) \cap S$. It then follows for all $p_0 \in \mathcal{U}_\epsilon(p) \cap S$ that*

$$\|h(p) - h(p_0)\| \leq |\lambda_{max}|\, d_g(p, p_0) \leq \beta\, C\, d_g(p, p_0), \tag{9}$$

where λ_{max} is the principle curvature with the largest absolute value for any point in $\mathcal{U}_\epsilon(p) \cap S$ and the constant $C > 0$ depends on the weights of the neural network.

Robustness via smoothing



$$\mathrm{softplus}_{\beta}(x) = \frac{1}{\beta}\log(1 + e^{\beta x})$$

Smoothing: Connections to SmoothGrad

Theorem 2 *For a one-layer neural network* $g(x) = relu(w^T x)$ *and its* β*-smoothed counterpart* $g_\beta(x) = softplus_\beta(w^T x)$*, it holds that*

$$\underbrace{\mathbb{E}_{\epsilon \sim p_\beta}\left[\nabla g(x - \epsilon)\right]}_{\text{SmoothGrad}} = \underbrace{\nabla g_{\frac{\beta}{\|w\|}}(x)}_{\beta\text{-smoothing}},$$

where $p_\beta(\epsilon) = \frac{\beta}{(e^{\beta\epsilon/2} + e^{-\beta\epsilon/2})^2}$.

$\epsilon_i \approx \mathcal{N}(0, \sigma)$ with variance $\sigma = \log(2)\frac{\sqrt{2\pi}}{\beta}$

Robustness experiments

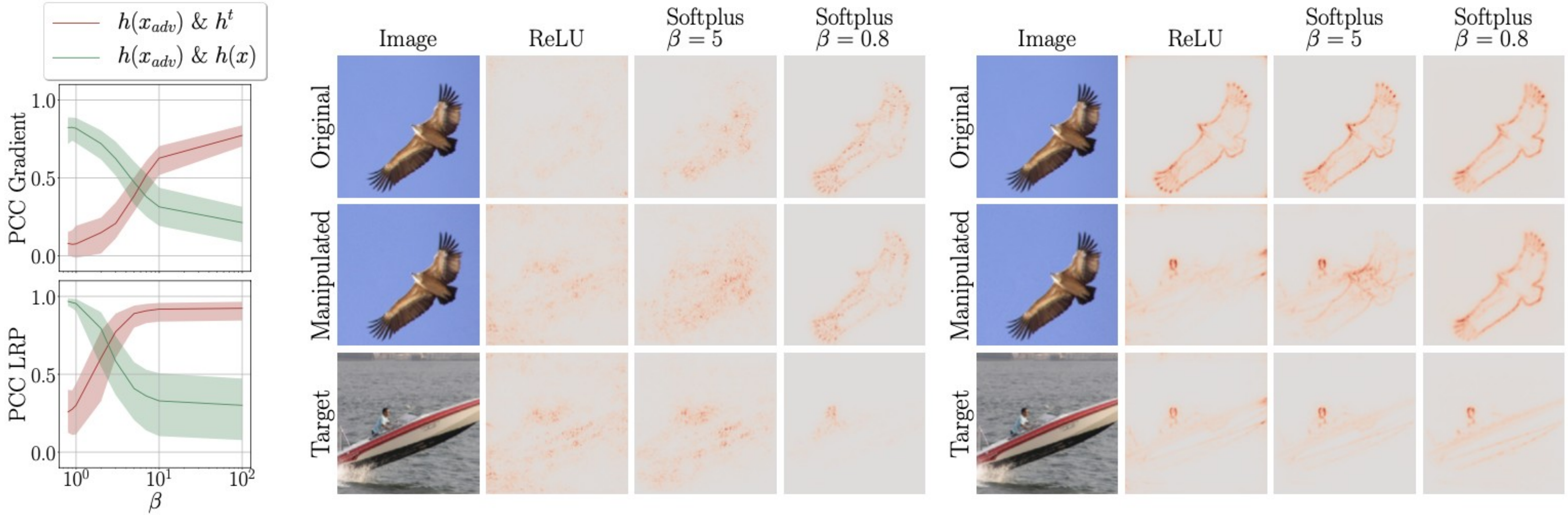


Figure 4. β-smoothing makes explanations more robust.

Robustness experiments

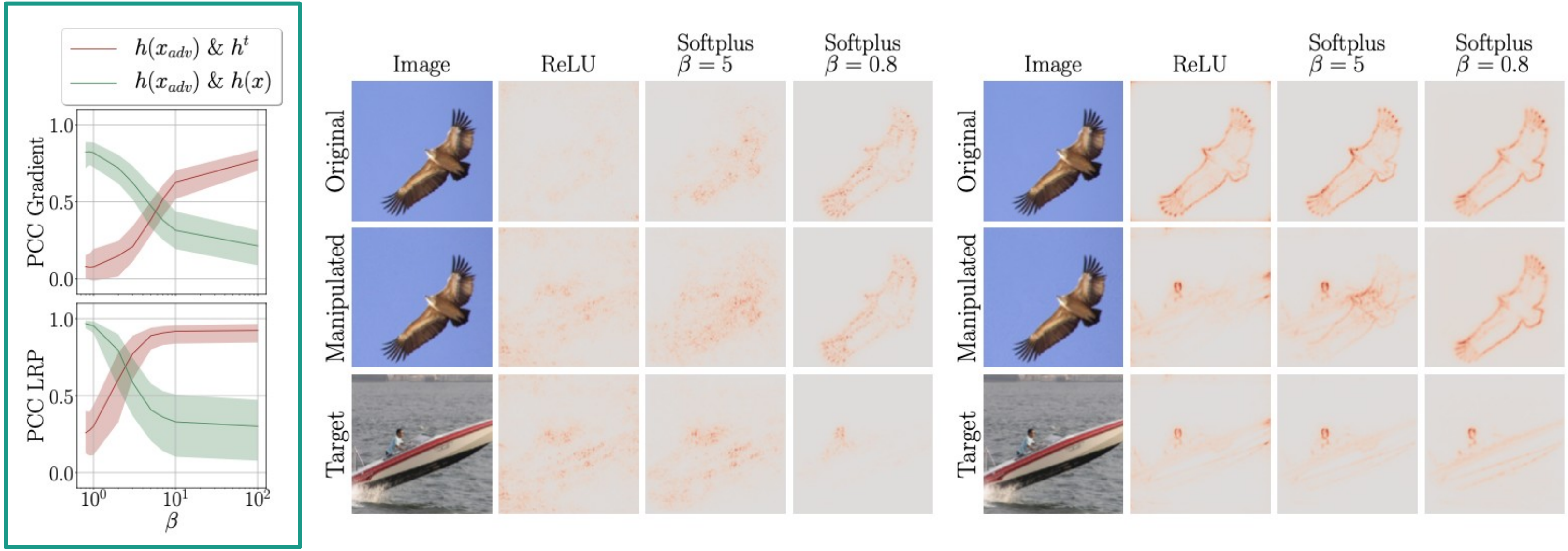


Figure 4. β-smoothing makes explanations more robust.

Robustness experiments

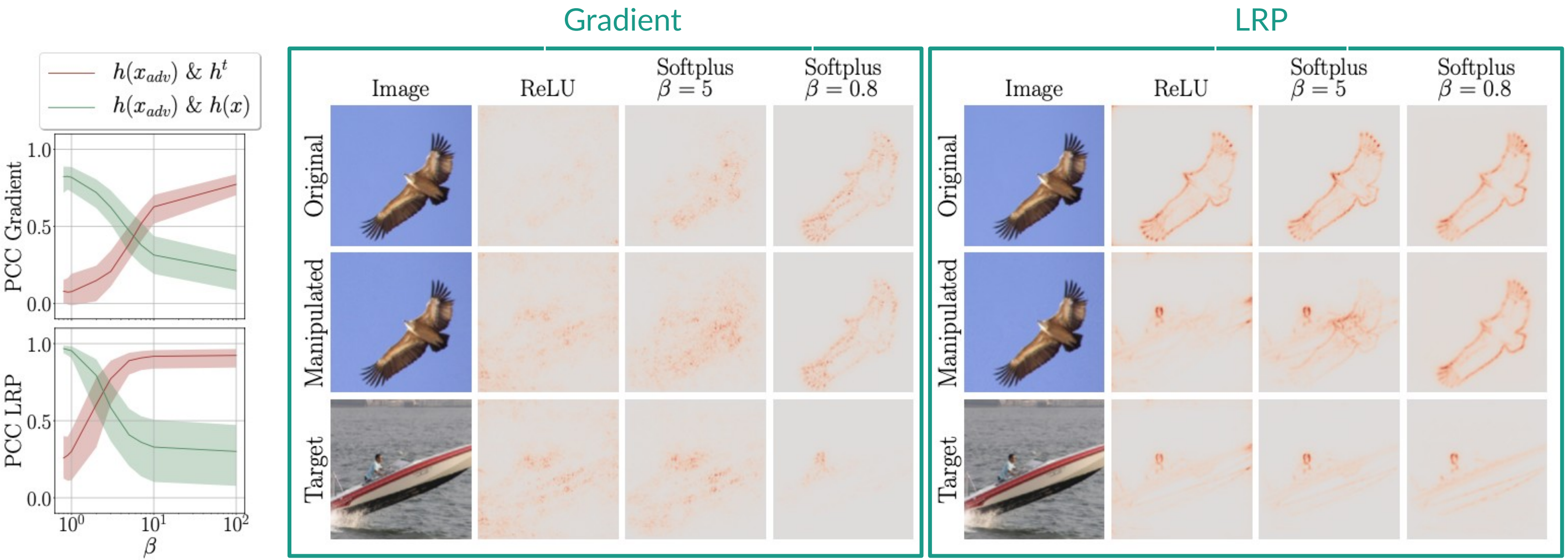


Figure 4. β-smoothing makes explanations more robust.

Robustness experiments

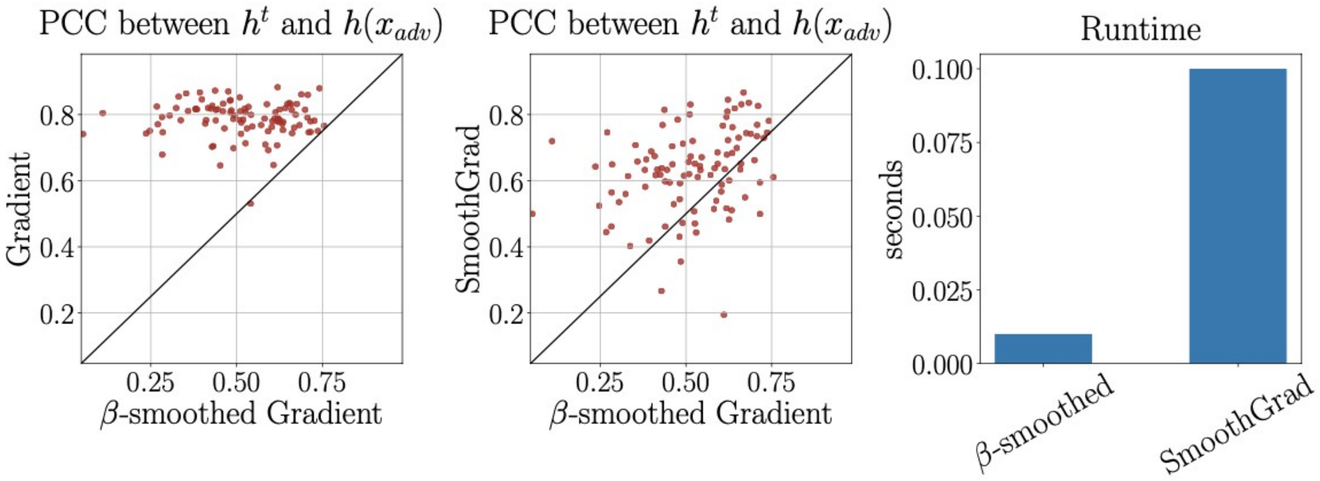


Figure 5. β-smoothing 1) makes explanations more robust
2) is comparable to SmoothGrad
3) has a faster runtime than SmoothGrad

Robustness experiments

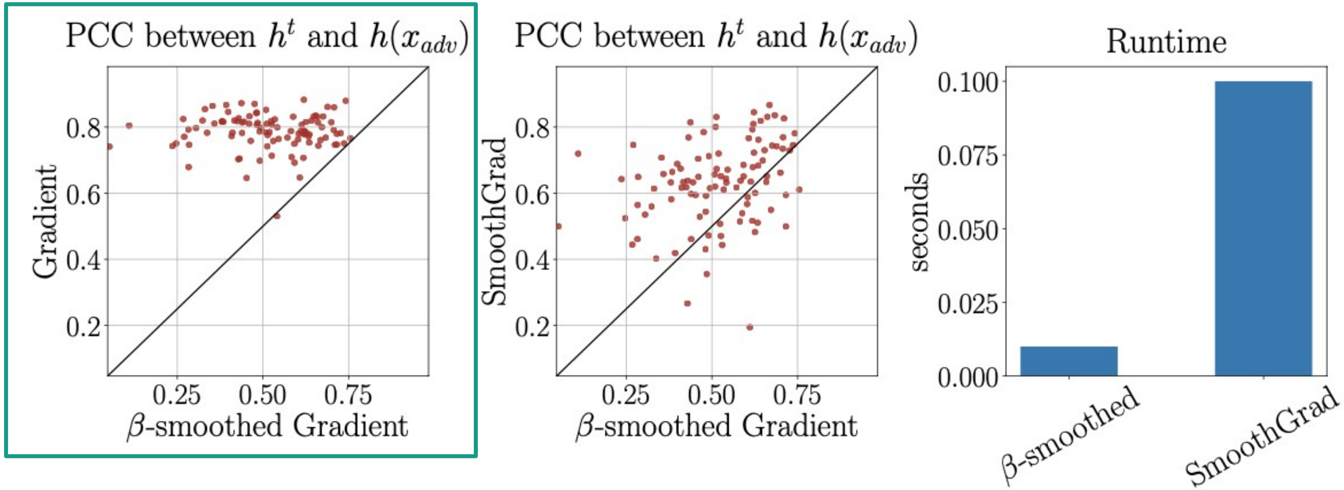


Figure 5. β-smoothing 1) makes explanations more robust
2) is comparable to SmoothGrad
3) has a faster runtime than SmoothGrad

Robustness experiments

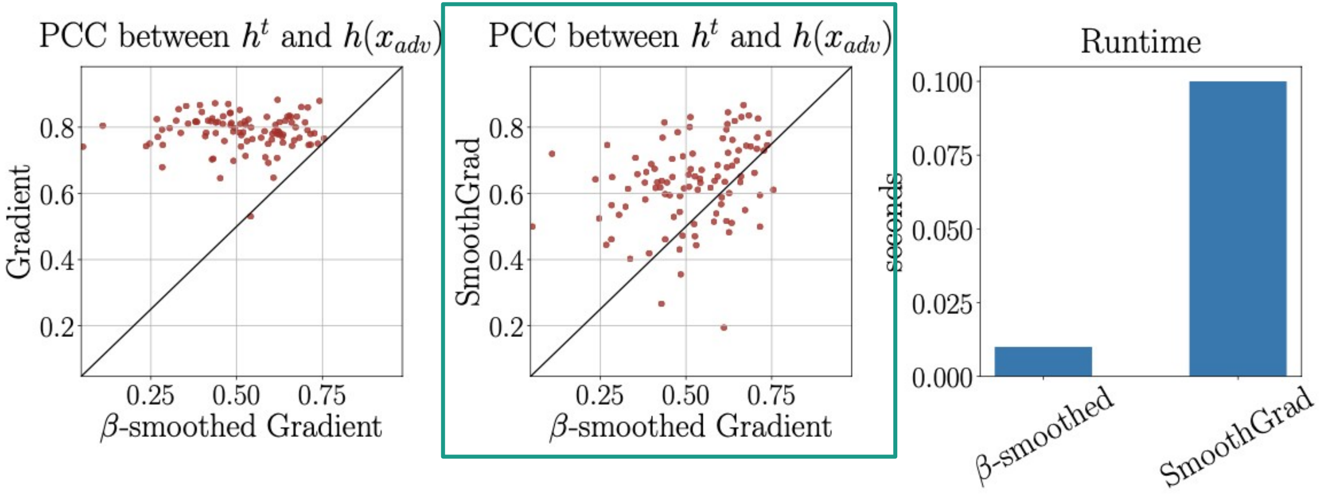


Figure 5. β-smoothing 1) makes explanations more robust
2) is comparable to SmoothGrad
3) has a faster runtime than SmoothGrad

Robustness experiments

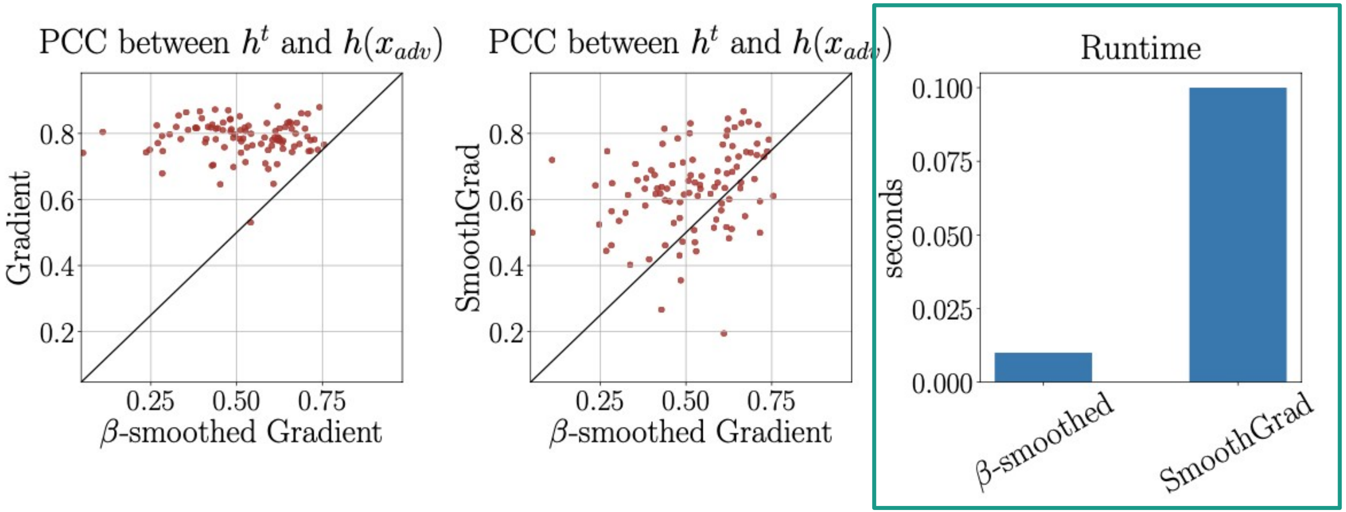


Figure 5. β-smoothing 1) makes explanations more robust
2) is comparable to SmoothGrad
3) has a faster runtime than SmoothGrad

Conclusion

Critique

Strengths

- Thorough investigation: problem → reason → solution
- Extensive validation: various explanation methods, models, and datasets

Limitations

- Analyses focused on relu/softplus activation function
- Evaluation of robustness based on Pearson correlation coefficient

Future directions & food for thought

Future directions

- Extend empirical analyses to other tasks and data modalities
- Generalize theoretical analyses to propagation-based methods
- Modify model training process to make NNs less vulnerable to explanation manipulation
    - Low-curvature models [Srinivas et al., NeurIPS 2022]

Food for thought

- Might there be other reasons for explanations being sensitive to manipulation?
- What are other ways to evaluate robustness of explanations?
- Is it a good idea to trade-off faithfulness for better robustness?