. SCUOLA INTERNAZIONALE SUPERIORE DI STUDI AVANZATI

SISSA Digital Library

Resilience of Bayesian Layer-Wise Explanations under Adversarial Attacks

Original
Resilience of Bayesian Layer-Wise Explanations under Adversarial Attacks / Carbone, G., Bortolussi, L., Sanguinetti, G.. - 2022:(2022), pp. 1-8. (2022 International Joint Conference on Neural Networks, IJCNN 2022 Padova, Italia 18-23 luglio 2022) [10.1109/IJCNN55064.2022.9892788].

Availability:
This version is available at: 20.500.11767/132270 since: 2023-05-11T12:19:33Z

Publisher:
Institute of Electrical and Electronics Engineers Inc.

Published
DOI:10.1109/IJCNN55064.2022.9892788

Terms of use:

Testo definito dall'ateneo relativo alle clausole di concessione d'uso

Publisher copyright

note finali coverpage

(Article begins on next page)

08 September 2026

Resilience of Bayesian Layer-Wise Explanations under Adversarial Attacks

1st Ginevra Carbone
Dept. of Mathematics and Geosciences, *University of Trieste*, Trieste, Italy
ginevra.carbone@phd.units.it

2nd Luca Bortolussi
Dept. of Mathematics and Geosciences, *University of Trieste*, Trieste, Italy
Modeling and Simulation Group, *Saarland University*, Saarland, Germany
luca.bortolussi@gmail.com

3rd Guido Sanguinetti
School of Informatics, *University of Edinburgh*, Edinburgh, United Kingdom
Machine Learning and Systems Biology group, SISSA, Trieste, Italy
gsanguin@sissa.it

arXiv:2102.11010v3 [cs.LG] 5 May 2022

***Abstract*—We consider the problem of the stability of saliency-based explanations of Neural Network predictions under adversarial attacks in a classification task. Saliency interpretations of deterministic Neural Networks are remarkably brittle even when the attacks fail, i.e. for attacks that do not change the classification label. We empirically show that interpretations provided by Bayesian Neural Networks are considerably more stable under adversarial perturbations of the inputs and even under direct attacks to the explanations. By leveraging recent results, we also provide a theoretical explanation of this result in terms of the geometry of the data manifold. Additionally, we discuss the stability of the interpretations of high level representations of the inputs in the internal layers of a Network. Our results demonstrate that Bayesian methods, in addition to being more robust to adversarial attacks, have the potential to provide more stable and interpretable assessments of Neural Network predictions.**

***Index Terms*—Adversarial attacks, Saliency explanations, Bayesian Neural Networks**

I. Introduction

Deep Neural Networks (DNNs) are the core engine of the modern AI revolution. Their universal approximation capabilities, coupled with advances in hardware and training algorithms, have resulted in remarkably strong predictive performance on a variety of applications, from computer vision to natural language to bioinformatics.

While the story of DNNs is indubitably one of success, it is tempered with a number of potentially very serious drawbacks which are somehow the natural flip side of dealing with extremely flexible and complex models. The first such drawback is the black box nature of DNNs: their expressivity and training on large data sets empirically results in very strong predictive power, but in general it does not provide any intuition about the possible explanations underlying the decisions. A second major drawback of DNN predictions is their vulnerability to adversarial attacks: empirically, it has been observed in many applications that well chosen infinitesimal changes in inputs can produce catastrophic changes in output [1], leading to paradoxical classifications and a clear problem in any application to safety critical systems. Such brittleness appears to be intimately related to the geometry of the data itself [2], and is therefore likely to be an intrinsic feature of standard DNN predictions.

In this paper, we argue theoretically and empirically [1] that these two problems are interlinked, and that therefore solutions that ameliorate resilience against adversarial attacks will also lead to more stable and reliable interpretations. We work within the framework of (pixel-wise) saliency explanations, which attempt to interpret post-hoc DNN decisions by apportioning a relevance score to each input feature for each data point. Specifically, we use the popular Layer-wise Relevance Propagation (LRP) [3], a method to assess the contribution of each pixel to the final classification score which backpropagates the prediction in the neural network until it reaches the input, using a set of suitable propagation rules. LRP saliency interpretations are well known to be unstable under perturbations of the inputs [4, 5, 6, 7]; recently, [8] suggested that a Bayesian treatment might ameliorate these stability problems.

Here, we consider the stability of saliency interpretations under targeted adversarial attacks that aim to change the classification under perturbations of the input. We introduce a novel notion of LRP robustness under adversarial attacks. As previously observed in [4, 9, 10], our results confirm that the LRP robustness of deterministic DNN predictions is remarkably low even when the adversarial attack fail to change the overall classification of the data point, i.e. that LRP interpretations are *less robust* than actual classifications. Considerations on the geometry of LRP [11] suggest that the observed lack of robustness might be imputable to large gradients of the prediction function in directions orthogonal to the data manifold. Here we expand on such a point of view, integrating it with a theoretical analysis in a suitably defined large-data limit [2, 12, 13, 14], and leveraging recent results from [2] about the robustness of BNNs to gradient based adversarial attacks. Specifically, we prove that Bayesian training of the DNNs in the large-data and overparameterized limit induces a regularizing effect which naturally builds robust explanations. We empirically validate this claim on the popular MNIST and Fashion MNIST benchmarks.

The main contributions of this paper are:

[1] Code is available at https://github.com/ginevracoal/BayesianRelevance.

1) The definition of a novel metric of robustness of interpretations of DNN results (Sec. IV-A);
2) A theoretical analysis of the effects of adversarial attacks on the original inputs, the effects of saliency attacks on prediction interpretations, and the improvements offered by a Bayesian treatment (Sec. IV-B);
3) An empirical study showing that indeed Bayesian training and prediction leads to more robust interpretations of classifications (Sec. V).

II. Background

A. Layer-Wise Relevance Propagation

Let $f : \mathbb{R}^d \to [0, 1]$ be a single-class image classifier, where $f(x, w)$ is the probability that an image $x \in \mathbb{R}^d$ belongs to the class of interest and w is the set of learnable weights. Without loss of generality, this concept can be extended to multi-class classifiers or to non probabilistic outputs. The idea of pixel-wise decomposition of a given image x is to understand how its pixels contribute to the prediction $f(x, w)$. In particular, LRP associates with each pixel p a relevance score $R(x_p, w)$, which is positive when the pixel contributes positively to the classification, negative when it contributes negatively to the classification and zero when it has no impact on the classification. All the relevance scores for a given image x can be stored in a heatmap $R(x, w) = \{R(x_p, w)\}_p$, whose values quantitatively explain not only whether pixels contribute to the decision, but also by which extent. One can leverage suitable propagation rules to ensure that the network output is fully redistributed through the network, namely that the relevance heatmap catches all the saliency features from the inputs [15]. For this purpose, the heatmap should be *conservative*, i.e. the sum of the assigned relevances should correspond to the total relevance detected by the model: $f(x, w) = \sum_p R(x_p, w)$. In the multi-label setting $R(x, w)$ is the heatmap associated to the classification label. Although the conservative property is not required to define a relevance heatmap, it has been empirically observed that conservative rules better support classification [15, 16]. Several propagation rules satisfy the conservative property, each of them leading to different relevance measures. In Sec. VII-D of the supplementary material we report three practical propagation rules: the Epsilon rule, the Gamma rule and the Alpha-Beta rule. [3] also presented LRP using a functional approach, i.e. independently of the network's topology. Then, [16] used *deep Taylor decomposition* to express any rule-based approach under the functional setting. Their method builds on the standard first-order Taylor expansion of a non-linear classifier at a chosen *root point* x^*, such that $f(x^*) = 0$,

$$\begin{aligned} f(x) &= f(x^*) + \nabla_x f(x^*) \cdot (x - x^*) + \gamma \\ &= \sum_p \frac{\partial f}{\partial x_p}\Big|_{x=x^*} \cdot (x_p - x_p^*) + \gamma, \end{aligned} \tag{1}$$

where γ denotes higher-order terms. The root point x^* represents a neutral image which is similar to x, but does not influence classification, i.e. whose relevance is everywhere null. The nearest root point to the original image x can be obtained by solving an iterative minimization problem [16]. The resulting LRP heatmap is $R(x) = \nabla_x f(x^*, w) \cdot (x - x^*)$.

B. Adversarial Attacks

Adversarial attacks are small perturbations of input data that lead to large changes in output (in the classification case, changes in predicted label) [1]. Broadly speaking, most attack strategies utilise information on the DNN loss function to detect an optimal perturbation direction, either through explicit knowledge of the loss function (*white box attacks*) or via querying the DNN (*black box attacks*, generally weaker than their white box counterpart).

White box attacks generally utilize gradient information on the loss function to determine the attack direction. One of the best known gradient-based attacks is the Fast Gradient Sign Method (FGSM) [1], a one-step untargeted attack (i.e., an attack strategy which is independent of the class of the attacked point). An iterative, improved version of FGSM is Projected Gradient Descent (PGD) [17]. We define FGSM and PGD in Section VII-B of the Appendix.

The evaluation of adversarial attacks can be either qualitative or quantitative. In the qualitative case, one simply observes whether an attack strategy with a certain strength is successful in switching the classification label of the given data point; in general, the evaluation then reports the fraction of successful attacks on the whole data set. A quantitative metric to evaluate network performances against adversarial attacks is provided by the notion of *softmax robustness* [2], which computes the softmax difference between original and adversarial predictions as $1 - ||f(x) - f(\tilde{x})||_\infty$ and is a number between zero (maximal fragility) and one (complete robustness) for every data point.

C. Bayesian Neural Networks

Bayesian models capture the inherent uncertainty intrinsic in any model by replacing individual models with ensembles weighted by probability laws. In the DNN setup, a Bayesian Neural Network (BNN) consists of an ensemble of DNNs whose weights are drawn from the posterior measure $p(w|D)$, where D denotes the available training data. To compute the posterior measure, one needs to first define a prior distribution $p(w)$ on the network's weights w; by Bayes' theorem, the posterior is then obtained by combining the prior and the likelihood $p(w|D) \propto p(D|w)p(w)$, where the likelihood term $p(D|w)$ quantifies the fit of the network with weights w to the available training data. Exact computation of the posterior distribution $p(w|D)$ is in general infeasible, thus one needs to resort to approximate inference methods for training Bayesian NNs, namely Hamiltonian Monte Carlo (HMC) [18] and Variational Inference (VI) [19]. We describe both techniques in Section VII-E of the appendix.

Once the posterior distribution has been computed, BNNs produce predictions through the *posterior predictive distribu-*

tion

$$p(f(x)|D) = \int_w p(f(x)|w)p(w|D)dw \simeq \sum_{w_i \sim p(w|D)} p(f(x)|w_i),$$

where $f(x)$ is the output value at a new point x and the equality represents the celebrated *Bayesian model averaging* procedure. In the BNN setting, adversarial attacks are crafted against the posterior predictive distribution $p(f(x)|D)$. For example, FGSM attack on an ensemble of N networks $f(\cdot, w_i)$ with weights drawn from $p(w|D)$ becomes

$$\tilde{x} = x + \delta \ \text{sgn}\Big(\mathbb{E}_{p(w|D)}\big[\nabla_x L(x,w)\big]\Big) \simeq x + \delta \ \text{sgn} \sum_{i=1}^{N} \nabla_x L(x, w_i) \qquad w_i \sim p(w|D).$$

In a similar way, it is possible to introduce a concept of *Bayesian explanations* for BNN predictions. As is clear from the definition(s) of LRP in Section II-A, the relevance score assigned to an input feature depends on the neural network weights. Therefore, in the Bayesian setting, the relevance becomes a random variable; we define as a Bayesian explanation the expected relevance of a feature under the posterior distribution of the weights.

Recent research shows that Bayesian neural networks are adversarially robust to gradient-based attacks in the overparameterized and infinite data limit [2, 12]. It is therefore of interest to investigate whether such robustness also extends to the learned explanations. To do so, we compare the explanations of deterministic NNs to that of Bayesian NNs against such attacks. Fig. 6 shows an example of failed FGSM attack for a deterministic network and a Bayesian network with the same architecture. Despite the fact that the attack did not manage to change the overall classification, we can see immediately a large difference between the deterministic LRP explanation of the original image, $R(x)$, and of the adversarial image, $R(\tilde{x})$ (top row). On the other hand, in the Bayesian case (bottom row), the saliency maps before and after the attack are essentially identical. In the next sections, we will provide both a theoretical explanation of this phenomenon, and systematically substantiate empirically the robustness of Bayesian explanations to adversarial attacks.

III. Related Work

Most of the recent explanation methods provide post hoc interpretations of black box classifier, which generate visual explanations of the decisions performed on single input samples [20, 21, 22]. Among the variety of available techniques, gradient-based attribution methods rely on gradient information provided by the models to produce the explanations. We briefly mention a few of them in what follows. [23] compute image-specific saliency maps using a single back-propagation step across the network for a chosen class. *Local Interpretable Model-agnostic Explanations* (LIME) [24] searches for the optimal explanation of a sample from a specified class of explanation models which are intrinsically interpretable [22]. The attribution method of *Integrated Gradients* [25] satisfies two specific axioms: *sensitivity* and *implementation invariance*. Sensitivity indicates that whenever an input and a baseline differ by a single feature and their predictions on that input are distinct, the attribution associated to the differing feature is non-zero. Two functionally equivalent networks satisfy implementation invariance if they associate identical attributions to the same input. Integrated gradients have been extended in several works [26, 27, 28]. *Deep learning important features* (DeepLIFT) [29] assigns attributions by comparing the activation of each neuron to a reference activation, i.e. the activation of a baseline input, which is task-dependent. It satisfies the sensivity axiom but breakes implementation invariance [25]. *Shapley additive explanations* (SHAP) [30] generalizes all the explanation functions that can be expressed as a linear combination of binary variables, including LIME and DeepLIFT.

Adversarial machine learning has been extensively studied from the perspective of interpretability. Many works focus on adversarial manipulations of the explanations [4, 9, 10], which consists in altering the explanations without changing model predictions. Interestingly, [31] notice that aggregations of multiple explanation methods are less vulnerable to adversarial attacks. It has been observed that there is a strong connection between adversarial robustness and explainability of Neural Networks [32]. For instance, [33] use an adversarial training [1, 34] to improve the interpretability of the representations, while [35] present a regularization technique based on gradient smoothing, which favours adversarial robustness and interpretability simultaneously. Adversarial training is one of the most effective defense methods against the attacks, but it has also been used to improve sensitivity of the explanations to input perturbations [36]. In contrast to the previous studies, we provide a new concept of robustness of the explanations, where we compare the interpretation of an input to that of an adversarial attack in terms of common "relevant" pixels (Sec. IV).

A recent work [37] investigates how the stability of the interpretations could be improved by adding stochasticity to the model weights. Their *NoiseGrad* method relies on a *tempered Bayes posterior* [38] and aggregates the explanations provided by independent samples to produce a Bayesian explanation. Our idea that Bayesian Neural Networks could provide more stable explanations, instead, is motivated by the adversarial robustness of BNNs to gradient-based attacks [2]. Hence, we focus on gradient-based attacks and also provide a formal proof for the stability of LRP explanations (Thm. IV.1), which holds w.l.o.g. for any other gradient-based attribution method. Furthermore, we empirically evaluate our findings using VI and HMC approximate inference methods (Sec. V).

IV. Methodology

A. LRP Robustness

We define the *k-LRP robustness* of relevance heatmaps to adversarial attacks and use this measure to assess how adversarial perturbations of the inputs affect the explanations.

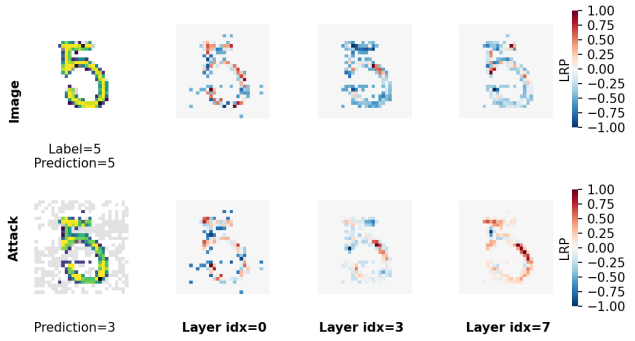


Fig. 1: LRP heatmaps of an image x and an FGSM adversarial perturbation $\tilde{x}$. Explanations are computed using the Epsilon rule w.r.t. the learnable layers of a deterministic network trained on the MNIST dataset. For each layer, we show the 30% most relevant pixels for both the original image and its adversarial counterpart, i.e. the ones selected for the computation of k-LRP$(x, \tilde{x}, w, l)$ from Eq. (2).

Definition IV.1. *Let x be an image with relevance heatmap $R(x, w)$ and $\tilde{x}$ an adversarial perturbation with relevance heatmap $R(\tilde{x}, w)$. Let Top$_k(R)$ denote the pixel indexes corresponding to the top k percent most relevant pixels in the absolute value of a heatmap R. The* k-LRP robustness *of x w.r.t. the attack $\tilde{x}$ is*

$$k\text{-}LRP(x, \tilde{x}, w) := \frac{|Top_k(R(x, w)) \cap Top_k(R(\tilde{x}, w))|}{k}. \quad (2)$$

In other words, the Top$_k(R)$ pixels have a strong positive or negative impact on classification and k-LRP$(x, \tilde{x}, w)$ is the fraction of common most relevant pixels for x and $\tilde{x}$ in the top k%. Fig. 5 in the Appendix gives an intuition of this computation. Notice that the LRP robustness of a point depends only implicitly on the strength of the attack through the attacked point $\tilde{x}$.

a) Inner layers explanations: We analyse the behaviour of LRP representations in the internal layers of the network, thus we also extend the computation of LRP heatmaps to any feature representation of the input x at a learnable layer $l \in \mathbb{N}$. We denote it by $R(x, w, l)$, where $l \leq L$ and L is the maximum number of layers available in the architecture. The corresponding LRP robustness will be denoted by k-LRP$(x, \tilde{x}, w, l)$. In such case, the robustness does not refer anymore to explanations in the classification phase (pre-softmax layer), but rather to the explanations in the learning phases, hence it gives an idea of the most relevant pixels determining an internal representation.

Fig. 1 shows an example of internal LRP heatmaps on a deterministic NN with learnable layers indexed by $l \in [0, 3, 7]$. For illustrative purposes, heatmaps appearing on the same row are normalized in $[-1, 1]$ before selecting the Top$_k$ pixels, since numeric scales are significantly different across the different internal representations.

b) Bayesian LRP robustness: The notion of LRP robustness can be naturally generalised to the Bayesian setting using the concept of Bayesian model averaging introduced in Section II-C. Hence, the LRP heatmap of a BNN is computed as the average of all the deterministic heatmaps from the ensemble: $\mathbb{E}_{p(w|D)}\big[k\text{-LRP}(x, \tilde{x}, w, l)\big]$. In this regard, we emphasise that Bayesian interpretations are affected by the chosen number of posterior samples drawn from the learned distribution.

B. Geometric meaning of adversarial interpretations

To better conceptualise the impact of a Bayesian treatment on LRP robustness, it is convenient to consider the thermodynamic limit of infinite data and infinite expressivity of the network, as formalised in [12, 13, 14]. For the purposes of our discussion, the main ingredients are the data manifold $\mathcal{M}_D$, a piecewise smooth submanifold of the input space where the data lie, and the true input/output function, which is assumed to be smooth and hence representable through an infinitely wide DNN. Practically, this limit might be well approximated on large data sets where the networks achieve high accuracy.

In this limit, it is proved [12, 13, 14] that the DNN $f(x, w)$ trained via SGD will converge to the true underlying function $g(x)$ over the whole data manifold $\mathcal{M}_D$. Because the data manifold is assumed to be piecewise smooth, it is possible to define a tangent space to the data manifold almost everywhere, and therefore to define two operators $\nabla_x^{\perp}$ and $\nabla_x^{\parallel}$ which define the gradient along the normal and tangent directions to the data manifold $\mathcal{M}_D$ at a point x of a function defined over the whole input space.

LRP and gradient-based adversarial strategies both share a reliance on gradient information. In the adversarial attacks cases, one evaluates the gradient of the loss function which, by the chain rule is given by

$$\nabla_x L(f, g) = \frac{\delta L(f, g)}{\delta f} \frac{\partial f}{\partial x} \quad (3)$$

In the thermodynamic limit, the DNN function $f(x, w)$ coincides with the true function everywhere on the data manifold, and therefore the tangent gradient of the loss function is identically zero. The normal gradient of the loss, however, is unconstrained by the data, and, particularly in a high dimensional setting, might achieve very high values along certain directions, creating therefore weaknesses that may be exploited by an adversarial attacker. The main result in [2] was to show that the orthogonal component of the loss gradient has expectation zero under the posterior weight distribution, therefore showing that BNNs are robust against adversarial attacks.

In the LRP setup, we instead consider gradients of the prediction function, as opposed to the loss, nevertheless the insight remains valid. The tangent components of the gradient of the prediction function $f(x, w)$ will coincide with the gradients of the true function $g(x)$, and therefore represent directions of true sensitivity of the decision function which are correctly recognised as relevant. However, such directions might be confounded or dwarfed by normal gradient components, which create directions of apparent relevance which, by construction, are targeted by gradient-based adversarial attacks. In Theorem

IV.1 we prove that BNNs in the thermodynamic limit will only retain relevant directions along the data manifold, which correspond to genuine directions of high relevance.

Theorem IV.1. *Let $\mathcal{M}_D \subset \mathbb{R}^d$ be an a.e. smooth data manifold and let $f(x, w)$ be an infinitely wide Bayesian neural network, trained on $\mathcal{M}_D$ and at full convergence of the training algorithm. Let $p(w|D)$ be the posterior weight distribution and suppose that the prior distribution $p(w)$ is uninformative. In the limit of infinite training data, for any $x \in \mathcal{M}_D$,*

$$\mathbb{E}_{p(w|D)}[\nabla_x^{\perp} f(x, w)] = 0.$$

Therefore, the orthogonal component of the gradient of the prediction function vanishes in expectation under the posterior weight distribution and Bayesian averaging of the relevance heatmaps naturally builds explanations in the tangent space $T_x\mathcal{M}_D$. Specifically, from the deep Taylor decomposition on a root point $x^* \in \mathcal{M}_D$ we obtain the expected LRP heatmap

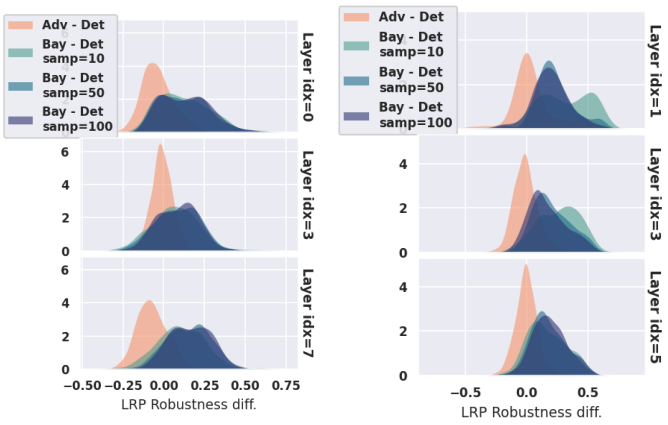
$$\mathbb{E}_{p(w|D)}[R(x)] = \mathbb{E}_{p(w|D)}\left[\nabla_x^{\parallel} f(x^*, w)\right] \cdot (x - x^*).$$

We refer the reader to Sec. VII-A of the Appendix for the theoretical background material and for a formal proof of Theorem IV.1. It should be noticed that LRP heatmaps at layer l involve partial derivatives w.r.t. x of the subnetwork $f^l(\cdot, w)$ of f, which associates to an input x the l-th activation from $f(\cdot, w)$. Consequently, the same vanishing property of the gradients holds for explanations in the internal layers - which are therefore more robust in the Bayesian case.

V. Experimental Results

In this section we corroborate the insights described in Section IV with an experimental evaluation, comparing empirically the LRP robustness using the popular MNIST [39] and Fashion MNIST [40] benchmark data sets. Both data sets are composed of 60.000 training images belonging to ten classes: in the MNIST case, these are hand-written digits, while the Fashion MNIST data set consists of stylized Zalando images of clothing items. While MNIST is considered a relatively trivial data set, with accuracies over 99% being regularly reported, Fashion MNIST is considerably more complex, and the best architectures report accuracies around 95%. We also extend the experiments to a ResNet architecture trained on 3-channels images from CIFAR-10 [41], which consists of 50.000 3-channels training images from ten classes. We do not examine more complex data sets, such as ImageNet [42], because of the very high computational costs of running Bayesian inference on very deep networks trained on very large data sets [2]. We train multiple DNNs and BNNs using both HMC and VI, which allows us to contrast the effect of a locally Gaussian approximation to the posterior against the asymptotically exact (but computationally more expensive) approximation provided by HMC. Because we require high accuracy in order to approximate the asymptotic conditions

[2]We do not experiment with scalable Monte Carlo dropout methods [43] here since there is no guarantee that their uncertainty estimates are able to capture the full posterior [44].



(a) (b)

Fig. 2: LRP robustness differences for FGSM (a) and PGD (b) attacks computed on 500 test points from Fashion MNIST dataset using the Epsilon rule. NNs in (a) have a convolutional architecture (Tab. I in the Appendix) and the BNN is trained with VI. NNs in (b) have a fully connected architecture (Tab. II in the Appendix) and the BNN is trained with HMC. BNNs are tested using an increasing number of samples $(10, 50, 100)$. Layer indexes refer to the learnable layers in the architectures.

described in Section IV-B, different architectures were used on the three data sets and between VI and HMC. In all cases, however, the BNN is compared with a DNN with the same architecture, to ensure fairness of the comparisons. Full details of the architectures used are reported in Sec. VII-G of the Appendix. Adversarial attacks in our tests are Fast Gradient Sign Method and Projected Gradient Descent, with a maximum perturbation size of 0.2. Saliency attacks are *target region* and *top-k* attacks [4] and *beta* attacks [10], which we briefly describe in Sec. VII-C of the Appendix. We rely on *TorchLRP* library [3] for the computation of LRP explanations and set $\epsilon = 0.1$ in the Epsilon rule, $\gamma = 0.1$ in the Gamma rule, $\alpha = 1$ and $\beta = 0$ in the Alpha-Beta rule. We report the computational resources in Sec. VII-F of the Appendix.

A. Bayesian interpretations are robust against the attacks

Our first significant result is that Bayesian explanations are more robust under attacks than deterministic architectures. For multiple data sets, attacks, training techniques (deterministic training, adversarial training, Bayesian inference) and approximate inference methods, LRP robustness scores are considerably higher than their deterministic counterparts. In Fig. 2 and Fig. 7,8,9 in the Appendix we show the distribution of pointwise differences w.r.t. the deterministic baselines between LRP robustness scores for MNIST and Fashion MNIST data sets using the Epsilon rule. The bottom row of the figures is the standard LRP (computed from the pre-softmax layer), while the top row is the initial feature representation (after the first non-linear layer), and the middle row is the LRP of an internal

[3]https://github.com/fhvilshoj/TorchLRP

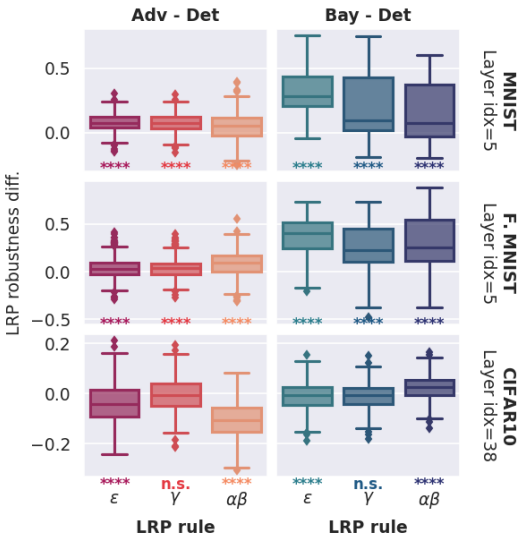
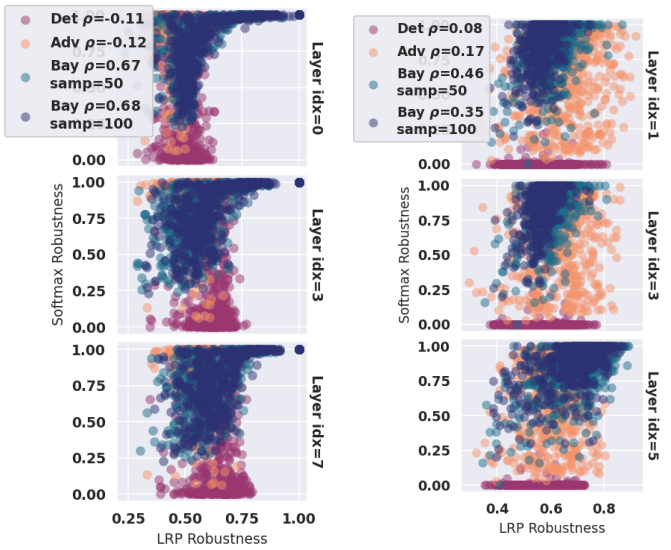


Fig. 3: LRP robustness differences for FGSM attack attacks computed on 500 test points from MNIST (top row), Fashion MNIST (middle) and CIFAR-10 datasets (bottom) using Epsilon, Gamma and Alpha-Beta rules (columns) on the Top_{20} pixels, for adversarially trained networks (left block) and BNNs (right block). NNs for MNIST and Fashion MNIST have a fully connected architecture (Tab. II) and the BNNs are trained with HMC. NNs for CIFAR-10 have a ResNet20 architecture from `bayesian_torch` library [45] and the BNN is trained with VI. BNNs are tested using 100 posterior samples. Layer indexes refer to the last learnable layer in each architecture. Parameters are described in the main text.



(a) (b)

Fig. 4: LRP vs softmax robustness of deterministic, adversarially trained and Bayesian NNs trained on MNIST dataset and tested against FGSM (a) and PGD (b) attacks. ρ denotes Spearman correlation coefficient. LRP Robustness is computed with the Epsilon rule on the 20% most relevant pixels. BNNs are trained with VI (a) and HMC (b) and are tested using an increasing number of samples $(50, 100)$. Layer indexes refer to the learnable layers in the convolutional (a) and fully connected (b) architectures (Tab. I and II in the Appendix).

layer. We tested Bayesian representations using an increasing number of posterior samples, i.e. $10, 50, 100$. We attacked 500 randomly selected test images, whose choice is balanced w.r.t. the available classes.

The first notable observation is that adversarially trained networks have low LRP robustness compared to BNNs: this confirms empirically the conjecture of Section IV-B that the components of the gradient that are normal to the data manifold (and are therefore the ones likely to be changed in an attack) are often major contributors to the relevance in DNN. On the contrary, the Bayesian averaging process greatly reduces the expected relevance of such direction.

In Fig. 3 and 10-13 in the supplementary material we test the stability of deterministic and Bayesian interpretations w.r.t. FGSM and PGD attacks on 500 test inputs from MNIST, Fashion MNIST and CIFAR-10 datasets. In Fig. 14 and Fig. 15 in the Appendix, we also test LRP stability against top-k, target region and beta saliency attacks (Sec. VII-C). As the saliency attacks are specifically designed to harm the interpretations, we consider them as proxy for the worst case robustness of the interpretations in a neighbor of the attacked point. Additionally, in Sec. VII-H of the Appendix we compare the robustness distributions for adversarially trained and Bayesian NNs using Mann-Whitney U rank test. We compute the relevance heatmaps using Gamma, Epsilon and Alpha-Beta rules. BNNs are trained with HMC and VI and evaluated using 100 samples from the posterior. The experiments confirm that Bayesian explanations are more stable across multiple LRP rules, gradient-based adversarial attacks and saliency attacks, also in the internal layers. Experiments on CIFAR-10 images in Fig. 3 (bottom row) with a ResNet architecture from `bayesian_torch` library [45] show only a modest improvement in adversarial and LRP robustness, probably due to the conditions of Thm IV.1 not being satisfied in this data set with higher complexity but relatively small size.

B. Bayesian LRP robustness increases with softmax robustness

A simple explanation for the improved LRP robustness of BNNs lies in the fact that BNNs are provably immune to gradient-based attacks [2]. Therefore, one might argue that the stability of the LRP is a trivial byproduct of the stability of the classifications.

To explore this question more in depth, we consider the relationship between the LRP robustness of a test point (stability of the explanation) and its softmax robustness (resilience of the classification against an attack). Fig. 4 and Fig. 16 in the supplementary material show scatterplots of these two quantities. An immediate observation is that deterministic explanations are weak against adversarial perturbations even when their softmax robustness is close to 1. Therefore, even in the cases

where the classification is unchanged, deterministic saliency heatmaps are fragile. In fact, there are no significant changes in LRP robustness between data points that are vulnerable to attacks and data points that are robust to attacks. Bayesian models, instead, show a strong positive correlation between LRP and softmax robustness, especially as the number of posterior samples increases. While it is immediately evident that Bayesian predictions are robust to adversarial attacks (since most data points have softmax robustness greater than 0.5), it is also clear from this correlation that attacks which are more successful (i.e. lower softmax robustness) also alter more substantially the interpretation, and are likely to represent genuine directions of change of the true underlying decision function along the data manifold.

VI. Conclusions

Deep neural networks are fundamental to modern AI, yet many of the mathematical structures underlying their success are still poorly understood. Unfortunately, an unavoidable consequence of this situation is that we also lack principled tools to address the weaknesses of deep learning systems. In this paper, we harness the geometric perspective to adversarial attacks introduced in [2] to study the resilience of Layer-wise Relevance Propagation heatmaps to adversarial attacks. The geometric analysis suggests a fundamental link between the fragility of DNNs against adversarial attacks and the difficulties in understanding their predictions: because of the unconstrained nature of classifiers defined on high dimensional input spaces but trained on low dimensional data, gradients of both the loss function and the prediction function tend to be dominated by directions which are orthogonal to the data manifold. These directions both give rise to adversarial attacks, and provide spurious explanations which are orthogonal to the natural parametrization of the data manifold. In the limit of infinite data, a Bayesian treatment remedies the situation by averaging out irrelevant gradient directions in expectation. Not only BNN interpretations are considerably more robust than deterministic DNN, but we also observe a correlation between softmax (adversarial) robustness and LRP robustness which suggests that indeed Bayesian interpretations capture the relevant parametrization of the data manifold.

We point out the presence of theoretical and practical limitations. The strong assumptions in Theorem IV.1, which restrict the geometrical considerations to fully trained BNNs in the limit of an infinite amount of weights and training data, do not prevent us from observing the desired behavior in practice, even when using cheap approximate inference techniques (VI). Indeed, Bayesian interpretations are considerably more robust than deterministic and adversarially trained networks. However, learning accurate BNNs on more complex datasets is extremely challenging, which makes the Bayesian scheme currently not suitable for large-scale applications. This suggests the need for further investigations on such matters, especially on sufficiently accurate and scalable approximate inference methods for BNNs [38]. Nevertheless, we believe that the insights provided by a geometric interpretation will be helpful towards a better understanding of both the strengths and the weaknesses of deep learning.

References

[1] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *ICLR*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[2] G. Carbone, M. Wicker, L. Laurenti, A. Patane, L. Bortolussi, and G. Sanguinetti, "Robustness of bayesian neural networks to gradient-based attacks," in *NeurIPS*, vol. 33. Curran Associates, Inc., 2020, pp. 15 602–15 613.

[3] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[4] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proceedings of AAAI*, vol. 33, no. 01, 2019, pp. 3681–3688.

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

[6] D. Alvarez-Melis and T. S. Jaakkola, "Towards robust interpretability with self-explaining neural networks," *arXiv preprint arXiv:1806.07538*, 2018.

[7] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, and T. Wang, "Interpretable deep learning under fire," in *29th USENIX*, 2020.

[8] K. Bykov, M. M.-C. Höhne, K.-R. Müller, S. Nakajima, and M. Kloft, "How much can i trust you?–quantifying uncertainties in explaining neural networks," *arXiv preprint arXiv:2006.09000*, 2020.

[9] J. Heo, S. Joo, and T. Moon, "Fooling neural network interpretations via adversarial model manipulation," *arXiv preprint arXiv:1902.02041*, 2019.

[10] A.-K. Dombrowski, M. Alber, C. J. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame," *arXiv preprint arXiv:1906.07983*, 2019.

[11] C. Anders, P. Pasliev, A.-K. Dombrowski, K.-R. Müller, and P. Kessel, "Fairwashing explanations with off-manifold detergent," in *ICML*. PMLR, 2020, pp. 314–323.

[12] G. M. Rotskoff and E. Vanden-Eijnden, "Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error," *stat*, vol. 1050, p. 22, 2018.

[13] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *ICML*. PMLR, 2019, pp. 1675–1685.

[14] S. Mei, A. Montanari, and P.-M. Nguyen, "A mean field view of the landscape of two-layer neural networks," *PNAS*, vol. 115, no. 33, pp. E7665–E7671, 2018.

[15] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2660–2673, 2016.

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

[17] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016.

[18] R. M. Neal *et al.*, "Mcmc using hamiltonian dynamics," *Handbook of markov chain monte carlo*, vol. 2, no. 11, p. 2, 2011.

[19] M. J. Wainwright and M. I. Jordan, *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

[20] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[21] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *CSUR*, vol. 51, no. 5, pp. 1–42, 2018.

[22] R. Marcinkevičs and J. E. Vogt, "Interpretability and explainability: A machine learning zoo mini-tour," *arXiv preprint arXiv:2012.01805*, 2020.

[23] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD*, 2016, pp. 1135–1144.

[25] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *ICML*. PMLR, 2017, pp. 3319–3328.

[26] K. Dhamdhere, M. Sundararajan, and Q. Yan, "How important is a neuron?" *arXiv preprint arXiv:1805.12233*, 2018.

[27] A. Jha, J. K. Aicher, M. R. Gazzara, D. Singh, and Y. Barash, "Enhanced integrated gradients: improving interpretability of deep learning models using splicing codes as a case study," *Genome biology*, vol. 21, no. 1, pp. 1–22, 2020.

[28] J. Merrill, G. Ward, S. Kamkar, J. Budzik, and D. Merrill, "Generalized integrated gradients: A practical method for explaining diverse ensembles," *arXiv preprint arXiv:1909.01869*, 2019.

[29] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *ICML*. PMLR, 2017, pp. 3145–3153.

[30] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st NeurIPS*, 2017, pp. 4768–4777.

[31] L. Rieger and L. K. Hansen, "A simple defense against adversarial attacks on heatmap explanations," *arXiv preprint arXiv:2007.06381*, 2020.

[32] R. Tomsett, A. Widdicombe, T. Xing, S. Chakraborty, S. Julier, P. Gurram, R. Rao, and M. Srivastava, "Why the failure? how adversarial examples can provide insights for interpretable machine learning," in *2018 21st FUSION*. IEEE, 2018, pp. 838–845.

[33] Y. Dong, H. Su, J. Zhu, and F. Bao, "Towards interpretable deep neural networks by leveraging adversarial examples," *arXiv preprint arXiv:1708.05493*, 2017.

[34] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[35] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Thirty-second AAAI*, 2018.

[36] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar, "On the (in) fidelity and sensitivity of explanations," *NeurIPS*, vol. 32, pp. 10 967–10 978, 2019.

[37] K. Bykov, A. Hedström, S. Nakajima, and M. M.-C. Höhne, "Noisegrad: enhancing explanations by introducing stochasticity to model weights," *arXiv preprint arXiv:2106.10185*, 2021.

[38] F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin, "How good is the bayes posterior in deep neural networks really?" *arXiv preprint arXiv:2002.02405*, 2020.

[39] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[40] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[41] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, vol. 55, no. 5, 2014.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[43] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*. PMLR, 2016, pp. 1050–1059.

[44] L. Smith and Y. Gal, "Understanding measures of uncertainty for adversarial example detection," *arXiv preprint arXiv:1803.08533*, 2018.

[45] R. Krishnan and P. Esposito, "Bayesian-torch: Bayesian neural network layers for uncertainty estimation," 2020.

[46] J. Lee, "Introduction to smooth manifolds (springer, 2012)." 2012.

[47] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: an overview," *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.

[48] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in practice*. CRC press, 1995.

[49] M. Betancourt, "A conceptual introduction to hamiltonian monte carlo," *arXiv preprint arXiv:1701.02434*, 2017.

[50] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[51] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.

[52] R. W. Hockney and J. W. Eastwood, *Computer simulation using particles*. crc Press, 2021.

VII. Appendix

A. Theoretical results

In the following, we summarize the main background material needed for proving Theorem IV.1.

Theorem 1 in [11] is a generalization of the submanifold extension lemma (e.g. Lemma 5.34 in [46]). It proves that any function g defined on a submanifold $\mathcal{M}$ can be extended to an embedding manifold $\mathcal{R}$, in such a way that the choice of the derivatives orthogonal to the submanifold is arbitrary. We change the original notation and report it here as Lemma VII.1.

Lemma VII.1. *Let $\mathcal{M} \subset \mathcal{R}$ be a k-dimensional submanifold embedded in the d-dimensional manifold $\mathcal{R}$. Let $V = \sum_{i=k+1}^{d} v^i \partial_i$ be a conservative vector field along $\mathcal{M}$ which assigns a vector in $T_x\mathcal{M}^\perp$ for each $x \in \mathcal{M}$. For any smooth function $g : \mathcal{M} \to \mathbb{R}$ there exists a smooth extension $F : \mathcal{R} \to \mathbb{R}$ such that*

$$F|_{\mathcal{M}} = g,$$

where $F|_{\mathcal{M}}$ denotes the restriction of F on the submanifold $\mathcal{M}$, and s.t. the derivative of the extension F is

$$\nabla_x F(x) = (\nabla_1 g(x), \dots, \nabla_k g(x), v^{k+1}(x), \dots, v^d(x))$$

for all $x \in \mathcal{M}$.

We use Lemma VII.1 and Corollary VII.1 to build an extension of the prediction function outside of the data manifold, with suitable derivatives in the orthogonal components.

In the proof of Corollary VII.1 we also leverage results on the global convergence of overparameterized neural networks [12, 13, 14] to approximate the extension of the prediction function with an infinitely wide neural network. Here we report the Universal Representation Theorem from [12] as Lemma VII.2, in the limit of infinite training data and infinite number of neurons/ weights (overparameterized network). In doing so, we refer to the notation in Sec. 7 of [2].

Lemma VII.2. *Let $f(x, w)$ be a neural network with differentiable and discriminating units, whose weights are drawn from a measure $\mu(w)$. Let $\tilde{f}(x)$ be a target function observed at points drawn from a data distribution $p(D)$ with support $\mathcal{M}_D \subset \mathbb{R}^d$. Suppose that:*

- *The input and feature spaces of f are closed Riemannian manifolds.*
- *The data distribution $p(D)$ is non degenerate.*

Then, the training loss function is a convex functional of the measure in the space of weights. Moreover, in the infinite data and overparameterized limit, stochastic gradient descent converges to the global minimum $\tilde{f}(x)$, i.e. the loss function is null on the data manifold $\mathcal{M}_D$.

Using Lemma VII.1 and Lemma VII.2, we can prove that it is always possible to select two sets of weights, w and w', such that the orthogonal gradients of the prediction function at a point x on the data manifold are opposite.

Corollary VII.1. *Let $\mathcal{M}_D \subset \mathbb{R}^d$ be an a.e. smooth manifold and let $f(x, w)$ be an infinitely wide Bayesian neural network, at full convergence of the training algorithm on $\mathcal{M}_D$. For any choice of weights w and $x \in \mathcal{M}_D$ there exists a set of weights w' such that $f(\cdot, w)|_{\mathcal{M}_D} = f(\cdot, w')|_{\mathcal{M}_D}$ and*

$$\nabla_x^\perp f(x, w) = -\nabla_x^\perp f(x, w'). \tag{4}$$

Proof. Let $F : \mathbb{R}^d \to \mathbb{R}$ be a smooth function s.t. $F = f(\cdot, w)$. From VII.1 we know that there exists a smooth extension F' of $F|_{\mathcal{M}_D}$ to the embedding space $\mathbb{R}^d$ such that $\nabla_x^\perp F'(x) = -\nabla_x^\perp f(x)$. Moreover, Lemma VII.2 holds for f in the limit of infinite weights, thus there exists a set of weights w' such that $f(\cdot, w') = F'$. This proves 4. □

Proof of Theorem IV.1. As a direct consequence of Corollary VII.1, under the assumption that the prior distribution $p(w)$ is uninformative, for any choice of weights such that the loss is zero on the data manifold, we obtain the same likelihood function and the same posterior distribution $p(w|D)$. This in particular holds for any pair w, w' constructed according to Corollary VII.1. Therefore, the orthogonal gradient of the prediction function evaluated on any $x \in \mathcal{M}_D$ vanishes in expectation, i.e.

$$\mathbb{E}_{p(w|D)}\left[\nabla_x^\perp f(x, w)\right] = 0.$$

□

B. Gradient-based Attacks

FGSM and PGD attacks are defined as follows:

- For a given network $f(\cdot, w)$ FGSM crafts a perturbation in the direction of the greatest loss w.r.t. the input

  $$\tilde{x} = x + \delta \ \text{sgn}\nabla_x L(x, w),$$

  where L is the training loss and δ is the perturbation magnitude (*strength* of the attack).
- PGD starts from a random perturbation in an ϵ-L_∞ ball around the input sample $\tilde{x}_0 := x$. At each step, it performs FSGM with a smaller step size, $\alpha < \epsilon$, and projects the adversary back in the ϵ-L_∞ ball

  $$\tilde{x}_{t+1} = \text{Proj}\{\tilde{x}_t + \alpha \cdot \text{sgn}\nabla_x L(\tilde{x}_t, w)\}_{\epsilon,\infty}.$$

  The size of the resulting perturbation is smaller than ϵ.

C. Saliency Attacks

In this section we define the adversarial perturbations of the explanations used in our experiments (Sec. V and Sec. VII-H), whose goal is to alter interpretations without affecting classifications.

[4] present a variety of attacks against feature importance methods, which iteratively maximize the diversity between original explanations $R(x)$ and perturbed explanations $R(\tilde{x})$. At each step the image is perturbed in the direction of the gradient of a *dissimilarity function* $D(x, \tilde{x})$

$$\tilde{x}_{t+1} = \text{Proj}\{\tilde{x}_t + \alpha \cdot \text{sgn}\nabla_x D(x, \tilde{x}_t)\}_{\epsilon,\infty}.$$

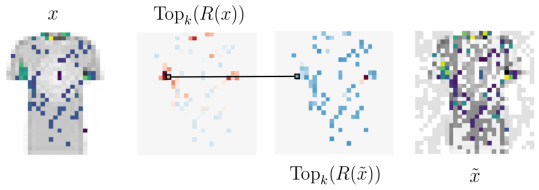


Fig. 5: Top_k pixels in an image x from Fashion MNIST dataset and an FGSM adversarial perturbation $\tilde{x}$.

We leverage two of the proposed techniques, with the following dissimilarity functions:

- $D(x, \tilde{x}) = \sum_{p \in \mathcal{A}_k} R(\tilde{x}_p)$ for the *target region* attack, where $\mathcal{A}_k$ is a pre-defined region of $k\%$ pixels.
- $D(x, \tilde{x}) = -\sum_{p \in \text{Top}_k(x)} R(\tilde{x}_p)$ for the *top-k* attack, where k indicates the chosen percentage of most relevant pixels.

In our experiments (Fig. 14) we set $\alpha = 0.5$, $k = 20$, number of iterations $T = 10$ and the region of pixels was chosen randomly.

For each test image x [10] build a targeted perturbation $\tilde{x}$ such that the classification is almost constant but the explanation resembles the target explanation of a randomly chosen test image $\hat{x} \neq x$. Specifically, they optimize the loss function

$$||R(\tilde{x}) - R(\hat{x})||^2 + \gamma \, ||f(\tilde{x}) - f(x)||^2$$

w.r.t. $\tilde{x}$ and clamp $\tilde{x}$ at each iteration to keep the image valid. Additionally, during the optimization phase they replace ReLU activations in the NN with softplus non-linearities

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

to avoid the problem of vanishing gradients. In our tests (Fig. 15 (b)) we set $\gamma = 1$, iterations $T = 10$ and learning rate $lr = 0.01$.

D. LRP Rules

A practical example of propagation rule is the *Epsilon rule* (ϵ-LRP) [47]. Let a_j be a neuron activation computed at the j-th layer during the forward pass. Let w_{jk} be the weight connecting a_j to a subsequent neuron a_k, w_{0k} be the bias weight and $a_0 = 1$. At each step of backpropagation LRP computes the relevance score R_j of the j-th activation a_j, by backpropagating the relevance scores of all neurons in the subsequent layer. The resulting ϵ-LRP score for a chosen $\epsilon > 0$ amounts to

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k.$$

The *Gamma rule* (γ-LRP) favours positive contributions over negative contributions by a factor of γ. The score for a chosen $\gamma > 0$ is

$$R_j = \sum_k \frac{a_j \cdot (w_{jk} + \gamma \max(0, w_{jk}))}{\sum_{0,j} a_j \cdot (w_{jk} + \gamma \max(0, w_{jk}))} R_k.$$

Finally, the *Alpha-Beta rule* ($\alpha\beta$-LRP) computes

$$R_j = \sum_k \left(\alpha \cdot \frac{\max(0, a_j w_{jk})}{\sum_{0,j} \max(0, a_j w_{jk})} - \beta \cdot \frac{\min(0, a_j w_{jk})}{\sum_{0,j} \min(0, a_j w_{jk})} \right) R_k,$$

where the conservative property holds for any choice of α and β s.t. $\alpha - \beta = 1$.

We refer to [47] (Sec 10.2.3) for a complete derivation of the propagation rules listed above within the Deep Taylor Decomposition framework [16]. Notice that for the $\alpha\beta$-LRP rule this generalization holds only when $\alpha = 1$ and $\beta = 0$, which are the values used in our experiments.

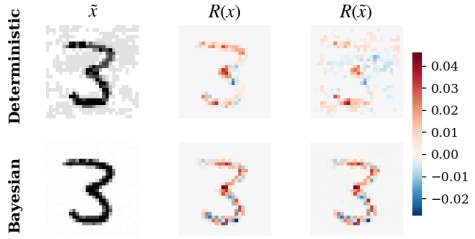


Fig. 6: LRP heatmaps of an image x (second column) and an FGSM adversarial perturbation $\tilde{x}$ (third column) which fails on a deterministic network and a Bayesian network. The two models have the same fully connected architecture and are both trained on the MNIST dataset. Explanations are computed using the Epsilon rule on the pre-softmax layer, i.e. layer idx $= 5$. Bayesian LRP is computed using 100 posterior samples. For each heatmap we only show the 30% most relevant pixels. The LRP robustness amounts to 0.46 in the deterministic case and to 0.68 in the Bayesian case.

E. Approximate Inference Methods

We briefly describe the methods used for performing approximate Bayesian inference in our experiments (Sec. V and VII-H).

Markov Chain Monte Carlo (MCMC) [48] is a class of stochastic algorithms that allow to sample from an unknown high-dimensional probability distribution (in our case the posterior $p(w|D)$) by building a Markov chain with the desired distribution as its equilibrium distribution. The efficiency of these methods in high-dimensional spaces is guaranteed by a proper exploration of the *typical set*, which is the region contributing the most to the computation of expectations w.r.t. the target density [49]. In the limit of infinite samples from the chain, the distribution of the recorded samples exactly matches the posterior distribution. *Hamiltonian Monte Carlo* (HMC) [18] is a Markov Chain Monte Carlo technique that combines an approximate Hamiltonian dynamics simulation and a *Metropolis-Hastings* [50, 51] acceptance step. It is designed to generate efficient transitions in the parameter space and to reduce the problems of low acceptance rates and

autocorrelation between consecutive samples. HMC introduces some momentum variables ρ and defines a Hamiltonian

$$H(\rho, w) = -\log p(\rho|w) - \log p(w) = T(\rho|w) + V(w),$$

where $p(\rho, w)$ is the joint density $p(\rho|w)p(w)$. Starting from the initial values of ρ and w, the system evolves according to Hamilton's equations

$$\frac{dw}{dt} = \frac{\partial H}{\partial \rho} = \frac{\partial V}{\partial \rho}$$
$$\frac{d\rho}{dt} = -\frac{\partial H}{\partial w} = -\frac{\partial T}{\partial w} - \frac{\partial V}{\partial w},$$

which are solved by means of *Leapfrog integration* [52]. At each step, Metropolis–Hastings computes the probability of keeping the new samples (ρ', w')

$$\min\{1, \exp\big(H(\rho, w) - H(\rho', w')\big)\}.$$

The second technique we use is *Variational Inference* (VI) [19]. It turns Bayesian inference into an optimization problem and provides an analytical approximation $q(w; \nu) \approx p(w|D)$ of the posterior distribution, namely the *variational distribution*, which belongs to a restricted family of known distributions (e.g. Gaussians). The variational parameters ν are optimized by minimizing the dissimilarity between p and q. The measure of similarity is the Kullback-Leiber divergence

$$\mathrm{KL}(q||p) := -\sum_{w} q(w; \nu) \log\Big(\frac{p(w|D)}{q(w; \nu)}\Big).$$

and the minimization problem

$$\nu^* = \arg\min_{\nu} \mathrm{KL}(q||p)$$

still requires the computation of the intractable term $p(w|D)$. Notice that the evidence $\log p(D)$ is constant with respect to ν and satisfies the equality

$$\log p(D) = \mathrm{KL}(q||p) + \mathbb{E}_q[\log p(w, D) - \log q(w; \nu)].$$

Therefore, the optimization problem above is equivalent to the minimization of the *Evidence Lower Bound* (ELBO) loss

$$\mathrm{ELBO}(\nu) = \mathbb{E}_q[\log p(w, D) - \log q(w; \nu)].$$

The first term in the ELBO loss encourages q to place its probability mass on the MAP estimate, while the second favours entropy on the mass, to avoid its concentration in a single location.

F. Computational Resources

Simulations were run on a machine with 36 cores, Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz processors and 192GB of RAM.

G. Architectures and Hyperparameters

TABLE I: Learnable layers and corresponding indexes in pytorch for the convolutional architecture. Hidden size is reported in Tab. III and Tab. IV.

Idx	Layer	Parameters
0	2D Conv.	`in_channels` = 784 `out_channels` = 32 `kernel_size` = 5
3	2D Conv.	`in_channels` = 32 `out_channels` = hidden size `kernel_size` = 5
7	F.c.	`in_features` = hidden size `out_features` = 10

TABLE II: Learnable layers and corresponding indexes in pytorch for the fully connected architecture. Hidden size is reported in Tab. III and Tab. IV.

Idx	Layer	Parameters
1	F. c.	`in_features` = 784 `out_features` = hidden size
3	F. c.	`in_features` = hidden size `out_features` = hidden size
5	F. c.	`in_features` = hidden size `out_features` = 10

TABLE III: Hyperparameters for BNNs trained with VI.

Dataset	MNIST	Fashion MNIST
Training inputs	60k	60k
Hidden size	512	1024
Nonlinear activations	Leaky ReLU	Leaky ReLU
Architecture	Convolutional	Convolutional
Training epochs	5	15
Learning rate	0.01	0.001

H. Additional Experiments

This section presents some more experiments performed on MNIST, Fashion MNIST and CIFAR-10 datasets, using BNNs trained with HMC and VI. Layer indexes refer to the learnable layers in each architecture. Parameters for computing the LRP heatmaps are set to: $\epsilon = 0.1, \gamma = 0.1, \alpha = 1, \beta = 0$.

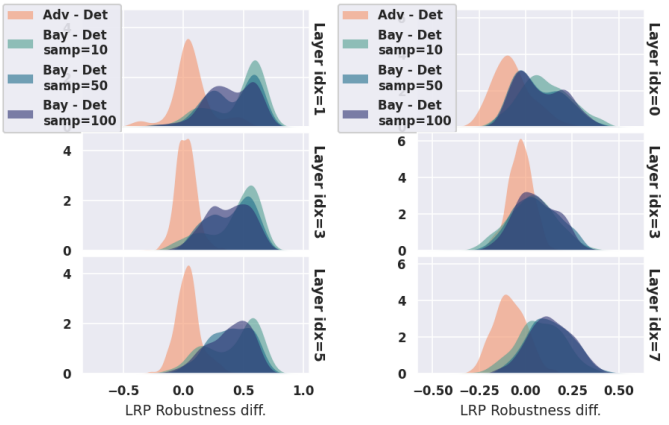
In Fig. 10-15 we also compare the robustness distributions for adversarially trained and Bayesian Neural Networks using Mann-Whitney U rank test. The asterisk notation (Tab. V) indicates statistically significant p-values in favour of the alternative hypothesis that the the distribution from the adversarially trained model is stochastically lower than the distribution from the Bayesian model. For significant p-values the Bayesian model is significantly more robust than the deterministic model.

TABLE IV: Hyperparameters for BNNs trained with HMC.

Dataset	**MNIST**	**Fashion MNIST**
Training inputs	60k	60k
Hidden size	512	1024
Nonlinear activation	Leaky ReLU	Leaky ReLU
Architectures	Fully connected	Fully connected
Warmup samples	100	100
Numerical integrator stepsize	0.5	0.5
Numerical integrator n. of steps	10	10

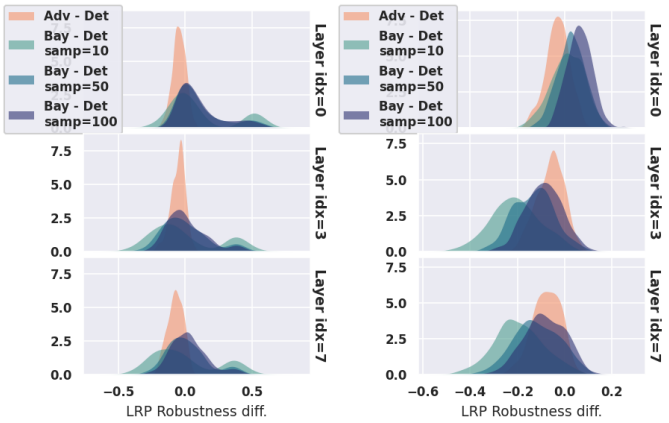
TABLE V: Asterisk notation for Mann-Whitney test.

p-value	**symbol**
$p > 0.05$	n.s.
$p \leq 0.05$	*
$p \leq 0.01$	**
$p \leq 0.001$	***
$p \leq 0.0001$	*****



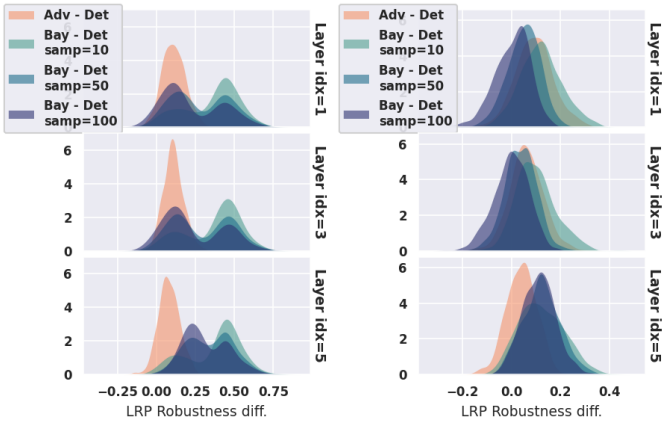
(a) (b)

Fig. 7: LRP robustness differences for FGSM (a) and PGD (b) attacks computed on 500 test points from Fashion MNIST dataset using the Epsilon rule. NNs in (a) have a fully connected architecture and the BNN is trained with HMC. NNs in (b) have a convolutional architecture and the BNN is trained with VI. BNNs are tested using an increasing number of samples $(10, 50, 100)$.



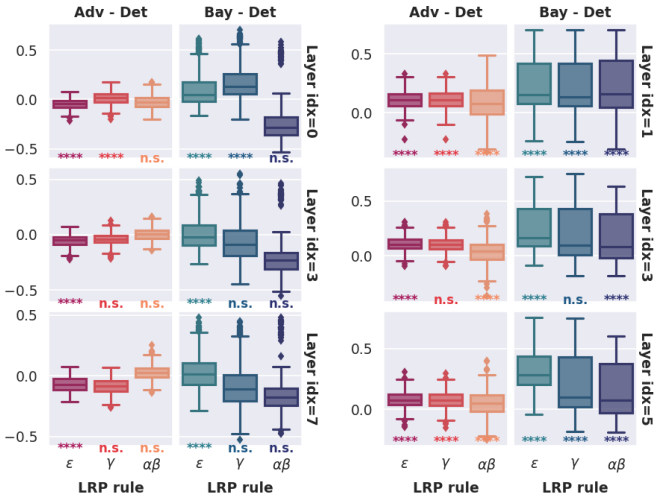
(a) (b)

Fig. 8: LRP robustness differences for FGSM (a) and PGD (b) attacks computed on 500 test points from MNIST dataset using the Epsilon rule. NNs have a convolutional architecture. BNNs are trained with VI and tested using an increasing number of samples $(10, 50, 100)$. Layer indexes refer to the learnable layers in the architectures.



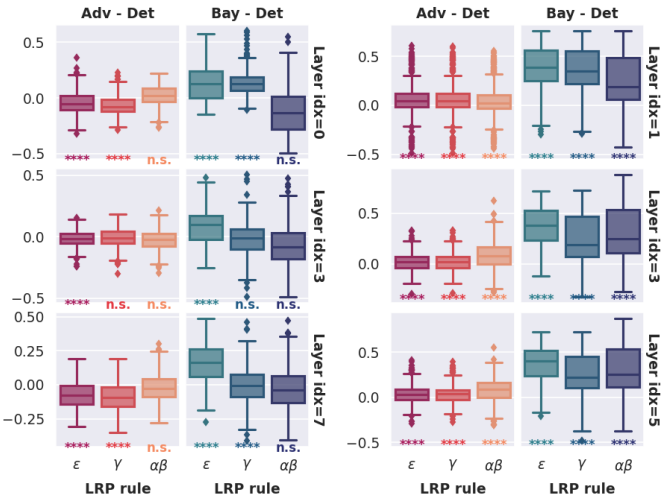
(a) (b)

Fig. 9: LRP robustness differences for FGSM (a) and PGD (b) attacks computed on 500 test points from MNIST dataset using the Epsilon rule. NNs have a fully connected architecture. BNNs are trained with HMC and tested using an increasing number of samples $(10, 50, 100)$.



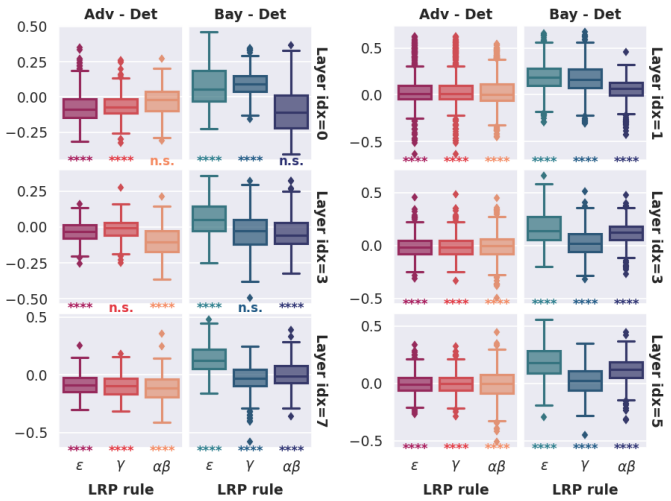
(a) (b)

Fig. 10: LRP robustness differences for FGSM attacks computed on 500 test points from MNIST dataset using Epsilon, Gamma and Alpha-Beta rules on the Top_{20} pixels. NNs in (a) have a convolutional architecture, while NNs in (b) have a fully connected architecture. The BNN in (a) is trained with VI and the BNN in (b) is trained with HMC; Both are tested using 100 posterior samples.



(a) (b)

Fig. 11: LRP robustness differences for FGSM attacks computed on 500 test points from Fashion MNIST dataset using Epsilon, Gamma and Alpha-Beta rules on the Top_{20} pixels. NNs in (a) have a convolutional architecture, while NNs in (b) have a fully connected architecture. The BNN in (a) is trained with VI and the BNN in (b) is trained with HMC; Both are tested using 100 posterior samples.



(a) (b)

Fig. 12: LRP robustness differences for PGD attacks computed on 500 test points from Fashion MNIST dataset using Epsilon, Gamma and Alpha-Beta rules on the Top_{20} pixels. NNs in (a) have a convolutional architecture, while NNs in (b) have a fully connected architecture. The BNN in (a) is trained with VI and the BNN in (b) is trained with HMC; Both are tested using 100 posterior samples.

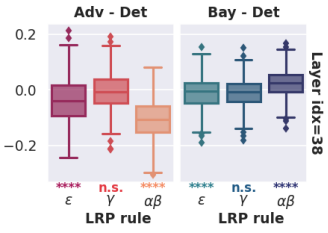
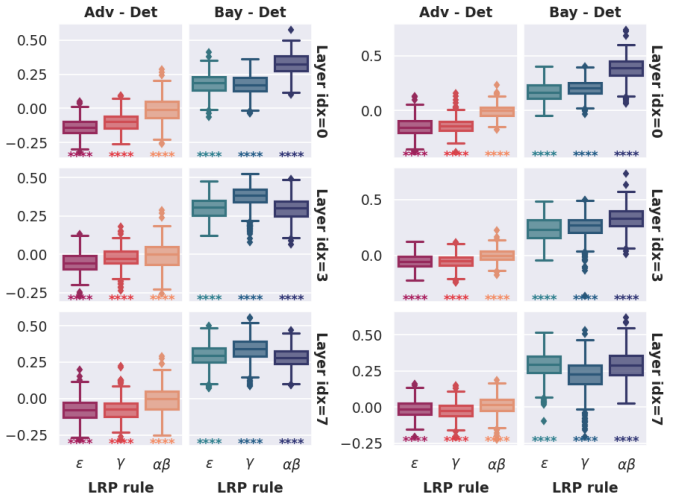


Fig. 13: LRP robustness differences for PGD attacks computed on 500 test points from CIFAR-10 dataset using Epsilon, Gamma and Alpha-Beta rules on the Top_{20} pixels. NNs have a ResNet20 architecture from `bayesian_torch` library [45]. The BNN is trained with VI and tested using 100 posterior samples.



(a) (b)

Fig. 14: LRP robustness differences for top-k (a) and target region (b) attacks [4] (Sec. VII-C) computed on 500 test points from MNIST (a) and Fashion MNIST (b) datasets using Epsilon, Gamma and Alpha-Beta rules on the Top_{20} pixels. NNs in (a) have a convolutional architecture, while NNs in (b) have a fully connected architecture. BNNs are trained with VI and tested using 100 posterior samples.

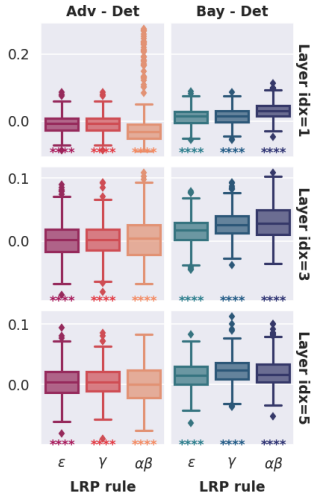
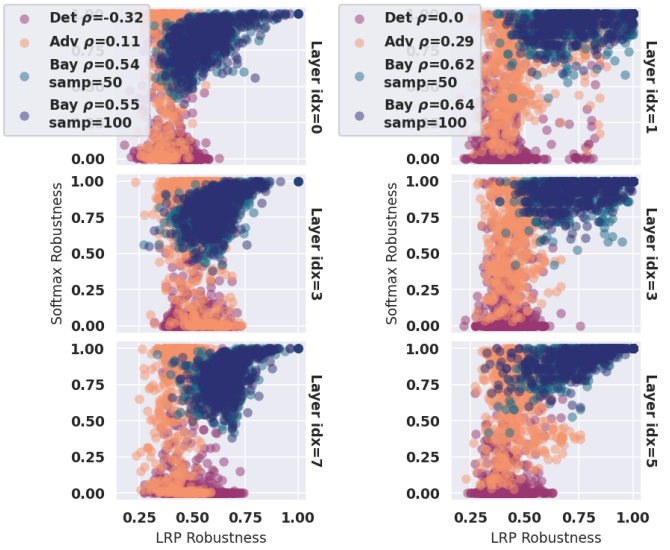


Fig. 15: LRP robustness differences for target beta attacks [10] (Sec. VII-C) computed on 500 test points from MNIST dataset using Epsilon, Gamma and Alpha-Beta rules on the Top_{60} pixels. NNs have a fully connected architecture (Tab. II in the Appendix). The BNN is trained with HMC and tested using 100 posterior samples.



(a) (b)

Fig. 16: LRP vs softmax robustness of deterministic, adversarially trained and Bayesian NNs trained on Fashion MNIST dataset and tested against FGSM attacks. ρ denotes the correlation coefficient. LRP Robustness is computed with the Epsilon rule on the 20% most relevant pixels. BNNs are trained with VI (a) and HMC (b) and are tested using an increasing number of samples $(50, 100)$.