Optimizing Explanations by Network Canonization and Hyperparameter Search

Frederik Pahde[1] Galip Ümit Yolcu[1,2] Alexander Binder[3,4]
Wojciech Samek[1,2,5,*] Sebastian Lapuschkin[1,*]
[1]Department of Artificial Intelligence, Fraunhofer Heinrich-Hertz-Institute, Germany
[2]Technische Universität Berlin, Germany
[3]ICT Cluster, Singapore Institute of Technology, Singapore
[4]University of Oslo, Norway
[5]BIFOLD – Berlin Institute for the Foundations of Learning and Data, Germany
∗ corresponding authors {wojciech.samek|sebastian.lapuschkin}@hhi.fraunhofer.de

Abstract

Explainable AI (XAI) is slowly becoming a key component for many AI applications. Rule-based and modified backpropagation XAI approaches however often face challenges when being applied to modern model architectures including innovative layer building blocks, which is caused by two reasons. Firstly, the high flexibility of rule-based XAI methods leads to numerous potential parameterizations. Secondly, many XAI methods break the implementation-invariance axiom because they struggle with certain model components, e.g., BatchNorm layers. The latter can be addressed with model canonization, which is the process of restructuring the model to disregard problematic components without changing the underlying function. While model canonization is straightforward for simple architectures (e.g., VGG, ResNet), it can be challenging for more complex and highly interconnected models (e.g., DenseNet). Moreover, there is only little quantifiable evidence that model canonization is beneficial for XAI. In this work, we propose canonizations for currently relevant model blocks applicable to popular deep neural network architectures, including VGG, ResNet, EfficientNet, DenseNets, as well as Relation Networks. We further suggest a XAI evaluation framework with which we quantify and compare the effects of model canonization for various XAI methods in image classification tasks on the Pascal VOC and ILSVRC2017 datasets, as well as for Visual Question Answering using CLEVR-XAI. Moreover, addressing the former issue outlined above, we demonstrate how our evaluation framework can be applied to perform hyperparameter search for XAI methods to optimize the quality of explanations. Code is available on `https://github.com/frederikpahde/xai-canonization`.

1. Introduction

In recent years, Machine Learning (ML) has been increasingly applied to high-stakes decision processes with a huge impact on human lives, such as medical applications [10, 35], credit scoring [45], criminal justice [47], and hiring decisions [9]. Therefore, awareness has been raised for the need of neural networks and their predictions to be transparent and explainable [14], which makes Explainable AI (XAI) a key component of modern ML systems. Rule-based and modified backpropagation-based XAI methods, such as DeepLift [36], Layer-wise Relevance Propagation (LRP) [5], and Excitation Backprop [48], that are among the most prominent XAI approaches due to their high faithfulness and efficiency, however, struggle when being applied to modern model architectures with innovative building blocks. This is caused by two problems: Firstly, rule-based XAI methods provide large flexibility thanks to configurable rules which can be tailored to the model architecture at hand. This comes at the cost of numerous potential XAI method parameterizations, particularly for complex model architectures. However, finding optimal parameters is barely researched and often neglected, which can cause these methods to yield suboptimal explanations. Secondly, earlier works [25] have shown that many XAI methods break implementation invariance, which has been defined as an axiom for explanations [42]. This is caused by certain layer types for which no explanation rules have been defined yet, e.g., BatchNorm (BN) layers. To address that issue, *model canonization* has been suggested, a method that fuses BN layers into neighboring linear layers without changing the underlying function of the model [15, 20], arguably leading to improved explanations for simple model architectures (VGG, ResNet) [29]. However, what constitutes a "good" explanation is only vaguely defined and many, partly contradicting, metrics for the quality of explanations have been proposed. Therefore, tuning hyperparameters of XAI methods and measuring the benefits of model canonization for XAI are non-trivial tasks.

To that end, we propose an evaluation framework, in which we evaluate XAI methods w.r.t their faithfulness, complexity, robustness, localization capabilities, and behavior with regard to randomized logits, following the authors of [17]. We apply our framework to (1) measure the impact of canonization and (2) demonstrate how hyperparameter search can improve the quality of explanations. Therefore, we first extend the model canonization approach to modern model architectures with high interconnectivity, e.g., DenseNet variants. We apply our evaluation framework to measure the benefits of model canonization for various image classification model architectures (VGG, ResNet, EfficientNet, DenseNet) using the ILSVRC2017 and Pascal VOC 2012 datasets, as well as for Visual Question Answering (VQA) with Relation Networks using the CLEVR-XAI dataset [4]. We show that generally model canonization is beneficial for all tested architectures, but depending on which aspect of explanation quality is measured, the impact of model canonization differs. Moreover, we demonstrate how our XAI evaluation framework can be leveraged for hyperparameter search to optimize the explanation quality from different points of view.

2. Related Work

2.1. XAI Methods

XAI methods can broadly be categorized into local and global explanations. While local explainers focus on explaining the model decisions on specific inputs, global explanation methods aim to explain the model behavior in general, e.g., by visualizing learned representations. For image classification tasks, local XAI methods assign relevance scores to each input unit, expressing how influential that unit (e.g., an input pixel) has been for the inference process. Many XAI methods are (modified) backpropagation approaches. To compute the importance of features in the detection of a certain class , they start from the output of the network, backpropagating importance values layer by layer, depending on the parameters and/or hidden activations of each layer. Saliency maps [6, 28, 38] are generated by computing the gradient $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$, where $f(\mathbf{x})$ is the model's prediction for an input sample $\mathbf{x}$. This yields a feature map, where each value indicates the model's sensitivity towards the corresponding feature. Guided Backpropagation [41] also uses the gradients, but applies the ReLU function to computed gradients in ReLU activation layers in the backpropagation pass. This filters out the flow of negative information, allowing to focus on the parts of the image where the desired class is detected. Integrated Gradients [42] accumulates the activation gradients on a straight path in the input space, starting from a baseline image $\mathbf{x}'$ selected beforehand, to the datapoint of interest. Formally, the attribution to the i^{th} feature is given by $(x_i - x'_i) \int_{\rho=0}^{1} \frac{\partial f(\mathbf{x}+\rho(\mathbf{x}'-\mathbf{x}))}{\partial x_i} d\rho$. SmoothGrad [40] aims to reduce the noise in saliency maps by sampling datapoints in the neighborhood of the original datapoint, and taking the average saliency map. Since these XAI methods rely only on the gradients of the total function computed by the network, they are implementation invariant, meaning they produce the same explanations for different implementations of the same function.

LRP [5] operates by redistributing the relevance scores of neurons backwards up to the input features. More precisely, LRP distributes the activation of the output neuron of interest to the previous layers in a way that preserves relevance across layers. Several rules have been defined (e.g., LRP-ϵ, LRP-γ, LRP-$\alpha\beta$), which can be combined in meaningful configurations according to the types and positions of layers in the neural network. Excitation Backprop [48] is a backpropagation method that is equivalent to LRP-$\alpha1\beta0$ [27], which has a probabilistic interpretation. DeepLIFT [37] is another rule-based method, where a reference image (e.g., the mean over the training population) is selected in addition. Using the associated rules, the differences in the activations of neurons on the reference image and the target image are backpropagated to the input. In addition to backpropagation-based XAI methods, there are also other approaches. Prominent examples are SHAP [24], which uses the game theoretic concept of Shapley values to find the contribution of each input feature to the model output, and LIME [31], which fits an interpretable model to the original model output around the given input. Both methods treat the model as a black box, only using outputs for certain inputs. As such, they are also implementation independent.

2.2. Evaluation of XAI Methods

While various XAI methods have been developed, the quantitative evaluation thereof is often neglected and explanations of XAI methods are often only compared by visually inspecting heatmaps. To address this issue, many XAI metrics have been introduced in recent years [1, 17]. However, there is no consensus on which metric to use and moreover, each metric evaluates explanations from different viewpoints, partly with contradictory objectives. Broadly speaking, XAI metrics can be categorized into five classes: **Faithfulness** metrics measure whether an explanation truly represents features used by the model. For instance, Pixel Flipping [5] measures the difference in output scores of the correct class, when replacing pixels in descending order of their relevance scores with a baseline value (e.g., black pixel or mean pixel). If the score decreases quickly, i.e., after replacing only a few highly relevant pixels, the explanation is considered as highly faithful. Region Perturbation [33] further generalizes Pixel Flipping by replacing input regions instead of single pixels. Faithfulness correlation [7] replaces a random subset of attribution with a baseline value and measures the correlation between the sum of attributions in the subset and the difference in model output. **Robustness** metrics measure the robustness of explanations towards small changes in the input. Prominent examples are Max-Sensitivity and Avg-Sensitivity [46], which use Monte Carlo sampling to measure the maximum and average sensitivity of an explanation for a given XAI method. **Localization** metrics measure how well an explanation localizes the object of interest for the underlying task. Consequently, in addition to the input sample and an explanation function, ground-truth localization annotations are required. Examples for localization metrics are Relevance Rank Accuracy (RRA) and Relevance Mass Accuracy (RMA) [4]. RRA measures the fraction of high-intensity relevances within the (binary) ground truth mask as RRA $= \frac{|P_{\text{top-K}} \cap GT|}{|GT|}$, where GT is the ground truth, K is the size of the ground truth mask and $P_{\text{top-K}}$ is the set of pixels sorted by relevance in decreasing order. Similarly, RMA measures the fraction of the total relevance mass within the ground truth mask and can be computed as RMA $= \frac{R_{\text{within}}}{R_{\text{total}}}$ where R_{within} is the sum of relevance scores for pixels within the ground truth mask and R_{total} is the sum of all relevance scores. **Complexity** metrics measure how concise explanations are. For example, the authors of [11] use the Gini Index of the total attribution vector to measure its sparseness, while [7] propose an entropy-derived measure. **Randomization** metrics measure by how much explanations change when randomizing model components. For instance, the random logit test [39] measures the distance between the original explanation and the explanation with respect to a random other class.

2.3. Challenges of Rule-Based/Modified Backpropagation Methods

No Implementation Invariance: From a functional perspective, it is desirable for an XAI method to be implementation invariant, i.e., the explanations for predictions of two different neural networks implementing the same mathematical function should always be identical [42]. However, rule-based and modified backpropagation approaches explain predictions from a message-passing point of view, which, by design, is affected by the structure of the predictor. This is impressively demonstrated by Montavon et al. [25], where the authors compute explanations for two different implementations of the same mathematical function and the relevance scores differ tremendously. Therefore, these methods violate the implementation invariance axiom, for example because of concatenations of linear operations such as BN and Convolutional layers. However, this problem can be overcome with model canonization, i.e., re-structuring the network into a canonical form implementing exactly the same mathematical function.

Parameterization: Rule-based backpropagation approaches are highly flexible and allow tailoring the XAI method to the underlying model and the task at hand. However, this flexibility comes at the cost of numerous possible parameterizations. For instance, the γ-rule in LRP computes relevances R_j of layer j given relevances R_k from the succeeding layer k as

$$R_j = \sum_k \frac{a_j \cdot (w_{jk} + \gamma w_{jk}^+)}{\sum_j a_j \cdot (w_{jk} + \gamma w_{jk}^+)} \cdot R_k \quad , \tag{1}$$

where a_j are the lower-layer activations, w_{jk} are the weights between layers j and k, w_{jk}^+ is the positive part of w_{jk} and γ is a parameter allowing to regulate the impact of positive and negative contributions. Therefore, γ is a hyperparameter that has to be defined for each layer. Note that the γ-rule becomes equivalent to the $\alpha 1 \beta 0$-rule as $\gamma \to \infty$, where negative contributions are disregarded. Similarly, for $\gamma = 0$, it is equivalent to the ϵ-rule, where negative and positive contributions are treated equally. The choice of γ for each layer can highly impact various measurable aspects of explanation quality.

3. Model Canonization

We assume there is a model f, which, given input data $\mathbf{x}$, implements the function $f(\mathbf{x})$. We further assume that f contains model components which pose challenges for the implementation of certain XAI methods. Model canonization aims to replace f by a model g where $g(\mathbf{x}) = f(\mathbf{x})$, but g does not contain the problematic components. We call g the canonical form of all models implementing the function $g(\mathbf{x})$. In practice, model canonization can be achieved by restructuring the model and combining several model components, as outlined in the following sections.

3.1. BatchNorm Layer Canonization

BN layers [21] were introduced to increase the stability of model training by normalizing the gradient flows in neural networks. Specifically, BN adjusts the mean and standard deviation as follows:

$$\text{BN}(\mathbf{x}) = w_{BN}^{\top} \left(\frac{\mathbf{x} - \mu}{\sqrt{\sigma + \epsilon}} \right) + b_{BN} \quad , \tag{2}$$

where w_{BN} and b_{BN} are learnable weights and a bias term of the BN layer, μ and σ are the running mean and running variance and ϵ is a stabilizer.

However, as discussed in Section 2.3, BN layers have shown to pose challenges for modified backpropagation XAI methods, such as LRP [20]. To address that problem, model canonization can be applied to remove BN layers without changing the output of the function. We make use of the fact that during test time the BN operation can be viewed as a fixed affine transformation. Specifically, we follow previous works [15], which have shown that BN layers can be fused with neighboring linear layers, including fully connected layers and Convolutional layers, of form $w_L^\top \mathbf{x} + b_L$, where w_L is the weight matrix and b_L is the bias term. This results in a single linear layer, combining the affine transformations from the original linear layer and the BN. The exact computation of the new parameters of the linear transformation depends on the order of model components:

Linear → BN: Many popular architectures (including VGG [43] and ResNets [16]) apply batch normalization directly after Convolutional layers. Hence, this model component implements the following function:

$$
\begin{aligned}
f(\mathbf{x}) &= \text{BN}(\text{Linear}(\mathbf{x})) && (3)\\
&= \underbrace{(\frac{w_{BN}}{\sqrt{\sigma+\epsilon}} w_L)}_{w_{\text{new}}}{}^\top \mathbf{x} + \underbrace{\frac{w_{BN}}{\sqrt{\sigma+\epsilon}}(b_L - \mu) + b_{BN}}_{b_{\text{new}}} && (4)
\end{aligned}
$$

which can be merged into a single linear layer with weight $w_{\text{new}} = \frac{w_{BN}}{\sqrt{\sigma+\epsilon}} w_L$ and bias $b_{\text{new}} = \frac{w_{BN}}{\sqrt{\sigma+\epsilon}}(b_L - \mu) + b_{BN}$. See Section A in the supplementary material for details.

BN → Linear: Other implementations apply BN right before linear layers (i.e., *after* the activation function of the previous layer), which impacts the computation of parameters of the merged linear transformation:

$$
\begin{aligned}
f(\mathbf{x}) &= \text{Linear}(\text{BN}(\mathbf{x})) && (5)\\
&= \underbrace{\frac{w_L^\top w_{BN}}{\sqrt{\sigma+\epsilon}}}_{w_{\text{new}}} \mathbf{x} \underbrace{- \frac{w_L^\top w_{BN}\mu}{\sqrt{\sigma+\epsilon}} + w_L^\top b_{BN} + b_L}_{b_{\text{new}}} && (6)
\end{aligned}
$$

Again, this component can be fused into a single linear transformation with weight $w_{\text{new}} = \frac{w_L^\top w_{BN}}{\sqrt{\sigma+\epsilon}}$ and bias $b_{\text{new}} = w_L^\top b_{BN} - \frac{w_L^\top w_{BN}\mu}{\sqrt{\sigma+\epsilon}} + b_L$. Note that there are practical challenges when padding is applied, for instance in a Convolutional layer. In this case, the bias becomes a spatially varying term, which cannot be implemented with standard Convolutional layers. See Section A in the supplementary material for details.

BN → ReLU → Linear: In some architectures, BN layers have to be merged with linear layers with an activation function (e.g., ReLU) in between. For instance, in DenseNets model components occur in that order. In that case, model canonization goes beyond merging two affine transformations, because of the non-linear activation function in between. Therefore, we propose to swap the BN layer and the activation function, which can be achieved by defining a new activation function, named *ReLU$_{thresh}$* which depends on the parameters of the **BN** layer, such that

$$
\text{ReLU}(\text{BN}(\mathbf{x})) = \text{BN}(\text{ReLU}_{\text{thresh}}(\mathbf{x})) \ , \qquad (7)
$$

where

$$
\text{ReLU}_{\text{thresh}}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } (w_{\text{BN}} > 0 \text{ and } \mathbf{x} > z) \\ \mathbf{x} & \text{if } (w_{\text{BN}} < 0 \text{ and } \mathbf{x} < -z) \\ z & \text{otherwise} \end{cases} \qquad (8)
$$

with $z = \mu - \frac{b_{\text{BN}}}{w_{\text{BN}}/\sqrt{\sigma+\epsilon}}$. Hence, **BN→ReLU→ Linear** is first transformed into **ThreshReLU→BN→ Linear**, then the BN layer and the linear layer can be merged with Eq. 6.

3.2. Canonization of Popular Architectures

We now demonstrate the canonization of popular neural network architectures. We picked 4 image classification models (VGG, ResNet, EfficientNet and DenseNet) and one VQA model (Relation Network [34]).

Image Classification Models: Many popular image classification model architectures, such as VGG [43], ResNet [16] and EfficientNet [44], apply BN directly after linear layers. Therefore, these networks can easily be canonized using Eq. 4. It gets more complicated, however, if model architectures are more complex with highly interconnected building blocks. DenseNets, for example, use skip connections to pass activations from each dense block to all subsequent blocks, as shown in Fig. 1. Each block applies BN on the concatenated inputs coming from multiple blocks, followed by ReLU activation and a Convolutional layer (BN → ReLU → Conv). Note that due to the high interconnectivity, BN layers cannot easily be merged into linear layers from neighboring blocks, because most linear layers pass their activations to multiple blocks, and vice versa, most blocks receive activations from multiple blocks. Consequently, the linear transformation implementing the BN function has to be merged with the linear layer following the ReLU activation within the same block. Therefore, we propose to perform model canonization by first applying Eq. 7 and then Eq. 6 to join BN layers with following linear layers within the same block, over the ReLU function between them. This process is visualized in Fig. 1. In addition, we apply Eq. 4 to merge the first BN layer in the initial layers of the DenseNet architecture, before the dense blocks. Moreover, there is a BN→ReLU→AvgPool2d→Conv chain in the end of the network, which can also be merged using Eq. 7 and Eq. 6.

VQA Model: In contrast to image classification models, VQA models, e.g., Relation Network [34], require two paths to encode both, the input image and the input question. Relation Networks use a simple Convolutional neural network as image encoder. The implementation by the authors of CLEVR-XAI [4] applies BN *after* the activation function (Conv → ReLU → BN). Therefore, we merge BN layers with the Convolutional layer from the following block using Eq. 6. The BN layer of the last block of the image encoder has to be merged with the fully connected

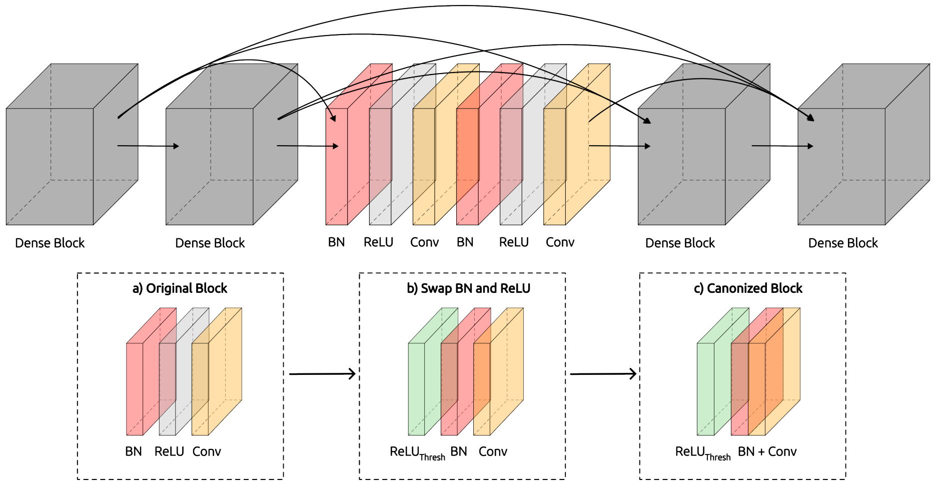

Figure 1. Due to the high interconnectivity of model components in DenseNets, it is not straightforward to fuse BN layers with Convolutional layers from neighboring blocks. Therefore, we suggest to first swap BN layers and ReLU activation functions using Eq. 7 (see step b)) and then merge the BN parameters into the following Convolutional layer using Eq. 6 (see step c)).

layer of the following module, which, however, receives a concatenation of image encoding and text encoding from the input question. Therefore, it has to be assured that BN parameters are merged only with the weights operating on inputs coming from the image encoder (see Section B and Fig. A.1 in the supplementary material for details).

4. Experiments: XAI Evaluation Framework

4.1. Datasets

ILSVRC2017 [32] is a popular benchmark dataset for object recognition tasks with 1.2 million samples categorized into 1,000 classes, out of which we randomly picked 50 classes for our experiments (see Section F.1 in the supplementary material). Bounding box annotations are provided for a subset of ILSVRC2017, which we use for localization metrics. Per class, we use up to 640 random samples. Note that ILSVRC2017 faces a center-bias, i.e., most of the objects to be classified are located in the center of the image. Therefore, naive explainers can assume that models base their decisions on center pixels. To that end, we include additional experiments using the Pascal VOC 2012 dataset [13] in Section D of the supplementary material.

CLEVR-XAI [4] builds upon the CLEVR dataset [22], which is an artificial VQA dataset. It contains 10,000 images showing objects with varying characteristics regarding shape, size, color and material. Moreover, there are simple and complex questions that need to be answered. The task is framed as a classification task, in which, given an image and a question, the model has to predict the correct response out of 28 possible answers. In total, there are approx. 40,000 simple questions, asking for certain characteristics of single objects. In addition, there are 100,000 complex questions, which require the understanding of relationships between multiple objects. CLEVR-XAI further comes with ground-truth explanations, encoded as binary masks locating the objects that are required in order to answer the question. Simple questions come with two binary masks, which are *GT Single Object*, localizing the object affected by the question, and *GT All Objects*, localizing all objects in the image. For complex questions, there are four binary masks, including *GT Union* localizing all objects that are required to answer the question (we refer to [4] for details on the other masks).

4.2. Models

For our experiments with ILSVRC2017, we analyze VGG-16 [43], ResNet-18 [16], EfficientNet-B0 [44] and DenseNet-121 [19]. We use pre-trained models provided in the PyTorch model zoo [30]. We use a Relation Network [34] for our experiments with CLEVR-XAI.

4.3. XAI Methods and Implementation Details

We analyze rule-based and modified backpropagation based XAI methods , namely Excitation Backprop (EB) and LRP. Note that other backpropagation-based methods, such as Saliency, Smoothgrad, Integrated Gradients and Guided Backprop are not impacted by model canonization [29] and are therefore not analyzed in this experiment. For each method, we compute explanations for both, the original and the canonized model. We use *zennit*[1] [2] as toolbox to compute explanations. For ILSVRC2017 with LRP, we analyze two pre-defined *composites*, i.e., mappings from layer type to LRP rule which have been established in literature [26], namely EpsilonPlus ($\epsilon+$) and Alpha2-Beta1

[1] https://github.com/chr5tphr/zennit

($\alpha2\beta1$), see Tab. A.1 in the supplementary material for details. For Relation Networks, we use a custom composite (*LRP-Custom*) following [4], in which we apply the $\alpha1\beta0$ rule to all linear layers and the box-rule [26] to the input layer. Note that ResNets, EfficientNets, and DenseNets leverage skip connections, which require the application of an additional canonizer in *zennit* to explicitly make them visible to the XAI method. Furthermore, we apply the signal-takes-it-all rule [3] to address the gate functions in the Squeeze-and-Excitation modules [18] in EfficientNets. In order to convert 3-dimensional relevance scores per voxel (channel $\times$ height $\times$ width) into 2-dimensional scores per pixel (height $\times$ width), we simply sum the relevances on the channel axis for ILSVRC2017 experiments. For CLEVR-XAI experiments, we follow the authors from [4] and use *pos-l2-norm-sq* ($R_{\text{pool}} = \sum_{i=1}^{C} max(0, R_i)^2$) as pooling function, where $C = 3$ is the number of channels. Results for the alternative pooling function *max-norm* ($R_{\text{pool}} = max(|R_1|, R_2, ..., R_C)$) are provided in the supplementary material. Moreover, before computing the metrics, we normalize the relevances by dividing all values by the square root of the second moment to bound their variance for numerical stability when comparing heatmaps.

4.4. XAI Metrics

In our experiments, we quantitatively measure the impact of canonization of the selected model architectures with various metrics, probing the quality of explanations from different viewpoints. We use the *quantus* toolbox[2] [17] to compute the following metrics: We measure **Faithfulness** using Region Perturbation with blurring as baseline function. We compute the Area over Perturbation Curve (AoPC) [33] to measure the faithfulness in a single number as AoPC $= \frac{1}{L+1}\left(\sum_{k=0}^{L} f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k)})\right)$, where $\mathbf{x}$ is the input sample, k is the perturbation step and L is the total number of perturbations. The AoPC is averaged over all input samples. **Localization** quality is measured using RRA and RMA. As ground truth location, we use bounding box annotations provided for ILSVRC2017, and binary segmentation masks for CLEVR-XAI. For the latter, we use *GT Unique* for simple questions and *GT Union* for complex questions. Moreover, we use the average sensitivity to measure **Robustness**, sparseness for **Complexity** and run the logit test as **Randomization** metric. While for robustness and randomization low scores are desirable, for the other metrics higher scores are better.

4.5. Canonization Results

The XAI evaluation results for the ILSVRC2017 dataset comparing models with and without canonization using the metrics described above are shown in Tab. 1. It can be seen that for most models and metrics the XAI methods yield better explanations for canonized models, especially for complexity, faithfulness, and localization metrics. However, there are some exceptions. While all other models yield less complex explanations when using model canonization, DenseNet explanations show the opposite behavior. This is due to the fact that many explanation heatmaps for DenseNets focus on small pixel groups (see Fig. A.7 in supplementary material) that, however, do not truly represent the model's behavior, as low faithfulness scores without canonization indicate. The localization metrics tend to be better for canonized models, except for EfficientNet-B0 with the $\alpha2\beta1$-composite for LRP, which, however, also yields a poor performance in terms of faithfulness. Hence, the $\alpha2\beta1$-composite itself appears to be a suboptimal parameterization for EfficientNets, which demonstrates the importance of the choice of hyperparameters for rule-based XAI methods. Note that randomization scores tend to increase for canonized models, i.e., the canonized model leads to explanations that are less dependent on the target class. This is due to the fact that the explanations are more focused on the object to classify (see improvements for localization metrics in Tab. 1) and therefore are more similar when computed for different target classes. Hence, randomization metrics have to be interpreted with caution [8]. Results for additional models and XAI evaluation metrics are shown in the supplementary material in Tables A.13-A.19.

In Tab. 2 we show results for our XAI evaluation using the CLEVR-XAI dataset. For LRP-Custom, model canonization yields explanations that are either better than those of the original model or approximately on par with them for both simple and complex questions, in particular for localization metrics. Results with *max-norm*-pooling are shown in the supplementary material in Tab. A.20.

4.6. XAI Hyperparameter Tuning

For our experiments in Section 4.5 we used pre-defined LRP composites established in literature [23, 26]. However, as suggested by different results for evaluated composites, these parameters differently impact the quality of explanations w.r.t the chosen metric. To that end, we run another experiment that uses our XAI evaluation framework for hyperparameter search. Specifically, we focus on the LRP-γ-rule, which uses the parameter γ to regulate the effect of positive and negative contributions (see Sec. 2.3), ranging from treating both equally ($\gamma = 0$) to neglecting negative contributions ($\gamma \rightarrow \infty$). Further, the flexibility of the LRP framework allows us to define XAI methods with varying focus on positive and negative contributions depending on the position of the layer in the network. We use a VGG-13 model with BN and define 4 groups of network layers, which are low-level (Conv 1-3), mid-level (Conv 4-7), and high-level

[2] https://github.com/understandable-machine-intelligence-lab/quantus

Table 1. XAI evaluation results with and without model canonization for VGG-16, ResNet-18, EfficientNet-B0 and DenseNet-121 using the ILSVRC2017 dataset. We measure the quality of explanations using the metrics Sparseness (*Complexity*), Region Perturbation (*Faithfulness*), RRA and RMA (*Localization*), Avg. Sensitivity (*Robustness*) and Random Logit Test (*Randomization*). Arrows indicate whether high (↑) or low (↓) are better. Best results are shown in bold.

Model	canonized	↑ Complexity no	↑ Complexity yes	↑ Faithfulness no	↑ Faithfulness yes	↑ Local. (RRA) no	↑ Local. (RRA) yes	↑ Local. (RMA) no	↑ Local. (RMA) yes	↓ Robustness no	↓ Robustness yes	↓ Random. no	↓ Random. yes
VGG-16	EB	0.57	**0.59**	0.35	**0.36**	0.70	**0.71**	0.68	**0.70**	0.22	**0.18**	1.00	1.00
	LRP-$\alpha 2 \beta 1$	0.70	**0.84**	0.38	**0.39**	0.63	**0.67**	0.65	**0.77**	**0.31**	0.34	**0.59**	0.66
	LRP-ε+	0.51	**0.62**	0.36	**0.39**	0.69	**0.71**	0.64	**0.71**	**0.19**	0.21	0.57	**0.54**
ResNet-18	EB	0.55	**0.57**	0.29	0.29	0.68	**0.69**	0.66	**0.67**	0.16	**0.14**	0.97	0.97
	LRP-$\alpha 2 \beta 1$	0.67	**0.76**	0.32	0.32	0.65	**0.67**	0.69	**0.75**	**0.21**	0.26	0.65	**0.61**
	LRP-ε+	0.51	**0.58**	0.30	0.30	0.69	**0.70**	0.65	**0.69**	**0.14**	0.15	0.70	0.70
EfficientNet-B0	EB	**0.85**	0.70	0.24	**0.27**	**0.73**	0.67	**0.79**	0.72	0.42	**0.33**	**0.99**	1.00
	LRP-$\alpha 2 \beta 1$	0.75	**0.77**	**0.29**	0.20	**0.72**	0.65	**0.79**	0.73	**0.48**	0.49	0.57	**0.51**
	LRP-ε+	0.50	**0.73**	0.28	**0.30**	0.75	0.75	0.69	**0.79**	**0.12**	0.21	**0.61**	0.65
DenseNet-121	EB	**0.66**	0.62	0.15	**0.31**	0.58	**0.72**	0.53	**0.73**	0.57	**0.17**	**0.75**	0.89
	LRP-$\alpha 2 \beta 1$	**0.82**	0.81	0.25	**0.33**	0.64	**0.71**	0.68	**0.81**	0.65	**0.28**	**0.40**	0.44
	LRP-ε+	**0.67**	0.66	0.26	**0.33**	0.70	**0.74**	0.71	**0.77**	0.63	**0.19**	**0.39**	0.48

Table 2. XAI evaluation results for Relation Network with and without model canonization using the CLEVR-XAI dataset for simple and complex questions using *pos-l2-norm-sq*-pooling. Arrows indicate whether high (↑) or low (↓) are better. Best results are shown in bold.

Questions	canonized	↑ Complexity no	↑ Complexity yes	↑ Faithfulness no	↑ Faithfulness yes	↑ Local. (RRA) no	↑ Local. (RRA) yes	↑ Local. (RMA) no	↑ Local. (RMA) yes	↓ Robustness no	↓ Robustness yes	↓ Random. no	↓ Random. yes
Simple	EB	**0.99**	0.97	0.50	**0.51**	**0.64**	0.61	**0.76**	0.70	**1.37**	1.39	1.00	1.0
	LRP-Custom [4]	0.95	**0.98**	0.52	0.52	0.70	0.70	0.75	**0.83**	**1.33**	1.35	**0.99**	1.0
Complex	EB	**0.99**	0.97	0.44	**0.45**	**0.66**	0.62	**0.82**	0.77	1.36	**1.35**	1.00	**0.99**
	LRP-Custom [4]	0.94	**0.97**	0.45	**0.46**	0.54	**0.63**	0.79	**0.86**	**1.33**	1.34	**0.98**	0.99

(Conv 8-10) layers, as well as fully-connected layers in the classification head. We define one γ-parameter per group with $\gamma \in \{0, 0.1, 0.25, 0.5, 1, 10\}$, and run a grid search for all possible combinations with and without model canonization, i.e., $2 \cdot 6^4 = 2592$ γ-configurations. Note that in theory, we could also evaluate different sub-canonizations, where we only canonize certain parts of the model. This, however, further increases the degrees of freedom. Further, note that more advanced multi-metric-objective hyperparameter optimization approaches can be employed. However, we decided to go forward with simple grid search, because our goal is to highlight the importance of the choice of XAI hyperparameters and the impact on various evaluation metrics. We evaluate the resulting explanations with the metrics described in section 4.4 and show the results in Fig. 2. Specifically, each line represents the score per metric with γ for a certain group of layers kept constant, averaged over all γ-parameterizations for other layer groups. It can be seen that the impact of the choice of γ depends on the position of the layer in the network and, in addition, differs by the metric of choice. For instance, the robustness of the explanations is mainly impacted by the γ-value in low-level layers (Conv 1-3), while it has no impact for the other layers. In contrast, randomization is mostly impacted by the choice of γ for fully connected layers in the classification head. Interestingly, canonization has a large impact on the optimal choice of γ for low-level layers when measuring the faithfulness of the resulting explanations. In Fig. 3 we show attribution heatmaps for three samples using different γ-configurations, employing the best and worst parameterization according to the metrics faithfulness, localization, and complexity. Each metric favors another parametrization, leading to different attribution heatmaps. High γ-values in low-level layers (γ_1) appear to be favorable for all metrics, i.e., more focus on positive contributions on those layers. This leads to attribution heatmaps with less noise, which is beneficial w.r.t faithfulness, localization, and complexity.

5. Conclusions

In this work, we proposed an evaluation framework for XAI methods which can be leveraged to optimize the quality of explanations based on a variety of XAI metrics. Specifically, we demonstrated the application of our framework to measure the impact of model canonization towards various aspects of explanation quality. Therefore, we extended the model canonization approach to state-

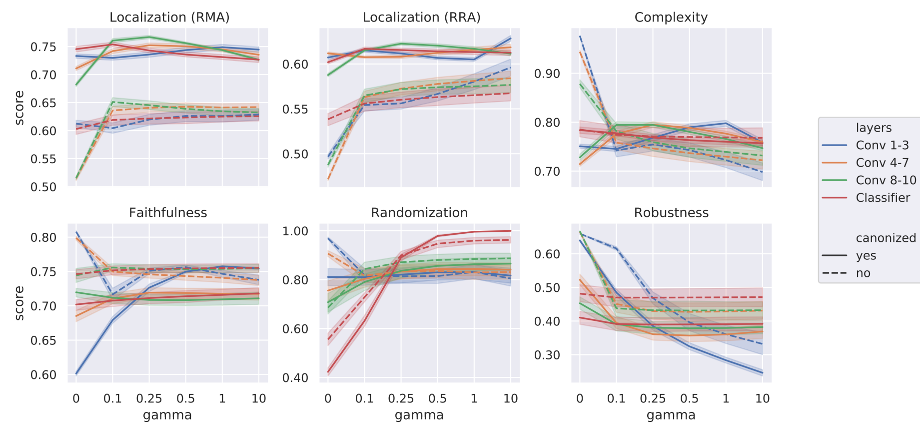

Figure 2. Results Grid Search for VGG-13: We group the layers into low-level (*Conv 1-3*), mid-level (*Conv 4-7*) and high-level (*Conv 8-10*) Convolutions, as well as fully connected layers in the classification head (*Classifier*). We evaluate different parameterizations for the γ-rule, where we define different values for γ per group and measure the quality of explanations both with and without model canonization w.r.t localization, faithfulness, complexity, randomization, and robustness metrics.

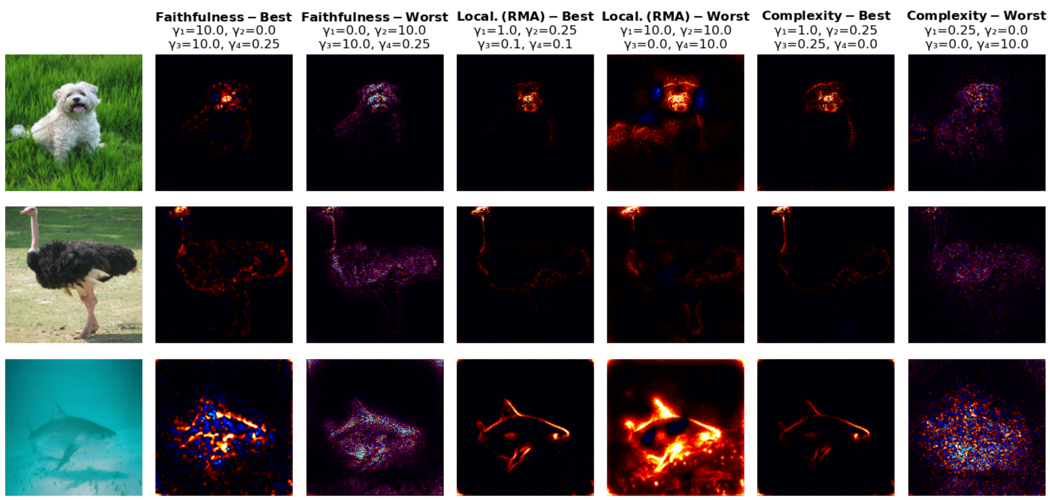

Figure 3. Attribution heatmaps (with canonization) for best and worst γ-parameters according to the grid search. γ_1 is for low-level features (Conv 1-4), γ_2 is for mid-level features (Conv 5-10), γ_3 is for high-level features (Conv 10-13), γ_4 is for layers in the classification head.

of-the-art model architectures, including EfficientNets and DenseNets. Despite not always being beneficial w.r.t. all examined architectures, model canonization provides an extra option when adopting XAI methods to the task at hand. Moreover, we applied our evaluation framework for hyperparameter optimization for XAI methods and demonstrated the impact of parameters w.r.t different XAI metrics. While we have evaluated our methods for LRP, it is also applicable to other configurable XAI methods, such as DeepLift. Future work will focus on the canonization of additional relevant model architectures, e.g., Vision Transformer [12]. In addition, optimizing the hyperparameter search is a promising research direction, e.g., with random search, evolutionary algorithms, or other approaches to reduce the search space. Moreover, the framework can be applied with other optimization objectives, e.g., to find LRP configurations that mimic other, more expensive XAI methods, e.g., SHAP.

Acknowledgements

This work was supported by the European Union's Horizon 2020 research and innovation programme (EU Horizon 2020) as grant [iToBoS (965221)]; the state of Berlin within the innovation support program ProFIT (IBB) as grant [BerDiBa (10174498)]; the Federal Ministry of Education and Research (BMBF) as grant [BIFOLD (01IS18025A, 01IS18037I)]; the German Research Foundation (ref. DFG KI-FOR 5363); the Research Council of Norway, via the SFI Visual Intelligence grant [project grant number 309439].

References

[1] Chirag Agarwal, Eshika Saxena, Satyapriya Krishna, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. Openxai: Towards a transparent evaluation of model explanations. *arXiv preprint arXiv:2206.11104*, 2022. 2

[2] Christopher J. Anders, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Software for Dataset-wide XAI: From Local Explanations to Global Insights with Zennit, CoRelAy, and ViRelAy. *CoRR*, abs/2106.13200, 2021. 5

[3] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. " What is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142, 2017. Publisher: Public Library of Science San Francisco, CA USA. 6

[4] Leila Arras, Ahmed Osman, and Wojciech Samek. CLEVR-XAI: a benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022. Publisher: Elsevier. 2, 3, 4, 5, 6

[5] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. Publisher: Public Library of Science San Francisco, CA USA. 1, 2

[6] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(61):1803–1831, 2010. 2

[7] Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and aggregating feature-based model explanations. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3016–3022. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track. 3

[8] Alexander Binder, Leander Weber, Sebastian Lapuschkin, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Shortcomings of top-down randomization-based sanity checks for evaluations of deep neural network explanations. *arXiv preprint arXiv:2211.12486*, 2022. 6

[9] Miranda Bogen and Aaron Rieke. Help wanted: An examination of hiring algorithms, equity, and bias. *Upturn, December*, 7, 2018. 1

[10] Titus J Brinker, Achim Hekler, Alexander H Enk, Carola Berking, Sebastian Haferkamp, Axel Hauschild, Michael Weichenthal, Joachim Klode, Dirk Schadendorf, Tim Holland-Letz, and others. Deep neural networks are superior to dermatologists in melanoma image classification. *European Journal of Cancer*, 119:11–17, 2019. Publisher: Elsevier. 1

[11] Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In *International Conference on Machine Learning*, pages 1383–1391. PMLR, 2020. 3

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arxiv 2020. *arXiv preprint arXiv:2010.11929*, 2010. 8

[13] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88:303–338, 2010. 5

[14] Bryce Goodman and Seth Flaxman. European Union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57, 2017. 1

[15] Mathilde Guillemot, Catherine Heusele, Rodolphe Korichi, Sylvianne Schnebert, and Liming Chen. Breaking batch normalization for better explainability of deep neural networks through layer-wise relevance propagation. *arXiv preprint arXiv:2002.11018*, 2020. 1, 4

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5

[17] Anna Hedström, Leander Weber, Daniel Krakowczyk, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. HÃ¶hne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34):1–11, 2023. 2, 6

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 6

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[20] Lucas YW Hui and Alexander Binder. Batchnorm decomposition for deep neural network interpretation. In *International Work-Conference on Artificial Neural Networks*, pages 280–291. Springer, 2019. 1, 3

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 3

[22] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 5

[23] Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. Towards best practice in explaining neural network decisions with LRP. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020. 6

[24] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 2

[25] Grégoire Montavon. Gradient-based vs. propagation-based explanations: An axiomatic comparison. In *Explainable ai: Interpreting, explaining and visualizing deep learning*, pages 253–265. Springer, 2019. 1, 3

[26] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019. Publisher: Springer. 5, 6

[27] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018. 2

[28] Niels JS Morch, Ulrik Kjems, Lars Kai Hansen, Claus Svarer, Ian Law, Benny Lautrup, Steve Strother, and Kelly Rehm. Visualization of neural networks using saliency maps. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 2085–2090. IEEE, 1995. 2

[29] Franz Motzkus, Leander Weber, and Sebastian Lapuschkin. Measurably stronger explanation reliability via model canonization. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 516–520, 2022. 1, 5

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and others. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[31] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 2

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[33] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016. Publisher: IEEE. 2, 6

[34] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30, 2017. 4, 5

[35] Farah Shahid, Aneela Zameer, and Muhammad Muneeb. Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. *Chaos, Solitons & Fractals*, 140:110212, 2020. Publisher: Elsevier. 1

[36] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 1

[37] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 2

[38] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[39] Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: Why many modified bp attributions fail. In *International Conference on Machine Learning*, pages 9046–9057. PMLR, 2020. 3

[40] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 2

[41] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 2

[42] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 1, 2, 3

[43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 4, 5

[44] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 4, 5

[45] Gang Wang, Jinxing Hao, Jian Ma, and Hongbing Jiang. A comparative assessment of ensemble learning for credit scoring. *Expert systems with applications*, 38(1):223–230, 2011. 1

[46] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[47] Aleš Završnik. Criminal justice, artificial intelligence systems, and human rights. In *ERA Forum*, volume 20, pages 567–583. Springer, 2020. 1

[48] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018. Publisher: Springer. 1, 2