TOWARDS A UNIFIED EVALUATION OF EXPLANATION METHODS WITHOUT GROUND TRUTH

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper proposes a set of criteria to evaluate the objectiveness of explanation methods of neural networks, which is crucial for the development of explainable AI, but it also presents significant challenges. The core challenge is that people usually cannot obtain ground-truth explanations of the neural network. To this end, we design four metrics to evaluate the explanation result without ground-truth explanations. Our metrics can be broadly applied to nine benchmark methods of interpreting neural networks, which provides new insights of explanation methods.

1 INTRODUCTION

Nowadays, many methods are proposed to explain the logic of a deep neural network (DNN) in a post-hoc manner. In this research, we limit our attention to existing methods of estimating the **importance/attribution/saliency** of input pixels or intermediate-layer neural units *w.r.t.* the network output (Shrikumar et al., 2016; Lundberg & Lee, 2017; Ribeiro et al., 2016; Binder et al., 2016), which present the mainstream of explaining neural networks. To avoid ambiguity, the estimated importance/saliency/attribution maps are all termed "attribution maps" in this paper.

However, some methods usually pursue attribution maps which look reasonable from the perspective of human users, instead of objectively reflecting the true logic of information processing in the DNN. A trustworthy evaluation of the objectiveness of attribution maps is crucial for the development of deep learning and proposes significant challenges to state-of-the-art algorithms.

Existing metrics (Cui et al., 2019; Arras et al., 2019; Vu et al., 2019; Yang & Kim, 2019; Kim et al., 2017; Adebayo et al., 2018; Ghorbani et al., 2019; Alvarez-Melis & Jaakkola, 2018) of evaluating explanation methods have certain shortcomings.

Issue 1, evaluation of the accuracy of a DNN $\neq$ evaluation of the objectiveness of attribution maps: *Some methods only evaluate whether a DNN encodes a correct logic, instead of examining whether an attribution map objectively reflects the true logic of a DNN.* (Cui et al., 2019) used human cognition to evaluate the explanation result. (Yang & Kim, 2019; Kim et al., 2017) aimed to construct a specific dataset with ground-truth explanations for evaluation. For example, they added an irrelevant object into the image. Pixels from the irrelevant object are expected to be assigned with zero attributions.

However, strictly speaking, it is impossible to religiously annotate ground-truth explanations for a DNN. Currently, the ground-truth explanation is constructed under the assumption that a DNN cannot learn irrelevant objects for classification with the purpose of evaluating the logic of the DNN, instead of examining whether an explanation method mistakenly generates attribution maps with seemingly correct logic for an incorrectly learned DNN.

Issue 2, broad applicability: We aim to design an evaluation metric that can be broadly applied to various tasks. In aforementioned methods (Yang & Kim, 2019; Kim et al., 2017), the requirement for constructing a new testing dataset limits the applicability of the evaluation.

Issue 3, quantification of the objectiveness: Some methods quantitatively evaluate the accuracy and robustness of attribution maps. However, there is no strict mechanism to ensure the objectiveness of each numerical value in the attribution map. *I.e.*, if the attribution value of a pixel is twice of that of another pixel, then the first pixel is supposed to contribute twice numerical values to the prediction *w.r.t.* the second pixel.

Table 1: Review of explanation methods

Method	What to explain	Quantitative evaluation of limitations in application
CAM (Zhou et al., 2016)	Attribution distribution at intermediate layer	1. Requirement for global average pooling. 2. Usually explain features at high layers
Grad_CAM (Selvaraju et al., 2017)	Attribution distribution at intermediate layer	Usually explain features at high layers
Grad (Simonyan et al., 2013)	Pixel-wise attribution	–
GI (Shrikumar et al., 2016)	Pixel-wise attribution	–
GB (Springenberg et al., 2014)	Pixel-wise attribution	Requirement for using ReLU as non-linear layers
Shapley Value (Shapley, 1953)	Pixel-wise attribution	NP-complete problem
DeepSHAP (Lundberg & Lee, 2017)	Pixel-wise attribution	Similar to LRP, DeepLIFT (Shrikumar et al., 2016) with a designed backward rule
LIME (Ribeiro et al., 2016)	Pixel-wise attribution	Attribution maps at the super-pixel level, rather than at the pixel level
LRP (Binder et al., 2016)	Pixel-wise attribution	Relevance propagation rules of every layer should be defined
Pert (Fong & Vedaldi, 2017)	Pixel-wise attribution	-

Except for the objectiveness, previous studies mainly conducted the evaluation from other perspectives. (Arras et al., 2019; Vu et al., 2019) evaluated attribution maps from the perspective of adversarial attacks by adding random noise. (Adebayo et al., 2018; Ghorbani et al., 2019; Alvarez-Melis & Jaakkola, 2018) proposed methods to evaluate the robustness of explanation methods *w.r.t.* the perturbation. (Adebayo et al., 2018) randomized the layer of DNN from the top to the bottom and visualized the change of attribution maps.

Note that in most applications, people cannot faithfully obtain the ground-truth logic of a DNN. Therefore, considering the above three issues, in this study, we aim to fairly evaluate the objectiveness and robustness of attribution maps from the following four perspectives without ground-truth explanations.

Perspective 1, bias of the attribution map at the pixel level: In order to evaluate the bias of the attribution map, we first need to propose a standard metric to evaluate the accuracy of explanation methods. The Shapley value is the unique solution to model the attribution value of each pixel that satisfies desirable properties including efficiency, symmetry and monotonicity (Lundberg & Lee, 2017). However, the computation of the Shapley value is an NP-complete problem, and previous studies (Lundberg & Lee, 2017) showed that the accurate estimation of the Shapley value is still a significant challenge. To this end, we extend the theory of Shapley sampling (Castro et al., 2009) and design a new evaluation metric, which achieves high accuracy without significantly boosting the computational cost.

We use the new evaluation metric to quantify the bias of the attribution map. Note that this evaluation has no partiality to the Shapley-value-based explanation methods. For example, experimental results showed that LRP (Binder et al., 2016) exhibited significant lower bias than DeepSHAP (Lundberg & Lee, 2017).

Perspective 2, quantification of unexplainable feature components: Given an input image and its attribution map, we revise the input image to generate a new image that reflects the logic of the attribution map. We then compare the intermediate-layer feature of the original image with that of the generated image, so as to disentangle feature components that can and cannot be explained by the attribution map.

Perspective 3, robustness of the explanation: Robustness of the explanation means whether the attribution map is robust to spatial masking of the input image. When we randomly mask a certain region of the input image, we admit that spatial masking destroys global contexts and affects pixel-wise attribution value to some extent. The quantification of the robustness of the explanation is an important perspective of evaluating an explanation method.

Perspective 4, mutual verification: The mutual verification means whether different explanation methods can verify each other. Methods generating similar attribution maps are usually believed more reliable.

In this paper, we used our metrics to evaluate nine widely used explanation methods listed in Table 1. We conducted experiments using the LeNet, VGG and ResNet on different benchmark datasets including the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset and the Pascal VOC 2012 (Everingham et al., 2010) dataset. Our experimental results proved the effectiveness of the proposed evaluation methods and provided an insightful understanding of various explanation methods.

The contribution of our work can be concluded as follows.
1. In this study, we invent a set of standard metrics to evaluate the objectiveness and the robustness of the attribution map without knowing ground-truth explanations.
2. The metric of evaluating the pixel-wise bias of the attribution map can be estimated with a relatively low computational cost, which avoids falling into the computational bottleneck of estimating accurate pixel-wise attributions.
3. Since our metrics do not need any annotations of ground-truth explanations, our metrics can be applied to different neural networks trained on different datasets.

2 Related Work

Explainable AI is an emerging direction in artificial intelligence, and different explanation methods have been proposed. In this section, we briefly review the Shapley value and limit our discussions to existing methods of evaluating methods of extracting attribution/importance/saliency maps to simplify the story. Appendix C discusses other research directions of explainable AI.

The Shapley value: The Shapley value (Shapley, 1953) was proposed to compute the attribution distribution over all players in a particular cooperative game. However, it is an NP-complete problem to compute the accurate Shapley value. The Shapley value approximated by sampling strategy could be very inaccurate due to the high variance. We extend the theory of the Shapley value to obtain an evaluation metric with a high accuracy but a low computational cost.

Qualitative evaluation: Some studies used a qualitative criterion for evaluation. (Cui et al., 2019) qualitatively defined basic concepts in the evaluation of explanation results, including the complexity of the explanation, the correlation, and the completeness. In contrast, this paper aims to evaluate the methods quantitatively, which makes our metrics more objective and reliable.

Accuracy evaluation: To evaluate the accuracy of attribution maps, (Arras et al., 2019; Vu et al., 2019) used the noise/occlusion to perturb the original image according to the attribution value. There was no mechanism to ensure the prediction result objectively reflected the logic of a DNN. (Yang & Kim, 2019; Kim et al., 2017) built a dataset to help them generate ground-truth explanations. Essentially, these methods tried to obtain the "correct" logic for an input image. However, a rigorous study should not assume that the DNN encodes the correct logic. As a result, this paper proposes to evaluate the objectiveness of explanation results without knowing ground-truth explanations.

Stability evaluation: (Adebayo et al., 2018; Ghorbani et al., 2019; Alvarez-Melis & Jaakkola, 2018) mainly paid attention to the attribution map change when the model input was perturbed. (Adebayo et al., 2018) visualized the change in the attribution map when the weights of the model were destroyed from the top to the bottom. (Ghorbani et al., 2019; Alvarez-Melis & Jaakkola, 2018) used the adversarial image to alter the attribution map. In comparison, we propose a metric to evaluate the robustness to spatial masking.

3 Algorithm

3.1 Preliminaries: The Shapley value

The Shapley value measures the instancewise feature importance ranking problem. Let Ω be the set of all pixels of an image I. I_P denotes an image that replaces all pixels in set $\Omega \setminus P$ with average pixel value over images. $F(I_P)$ denotes the scalar output of a DNN based on a subset of pixels $P \subset \Omega$. To compute the Shapley value of the i-th feature, (Shapley, 1953) considered all subsets of Ω not containing the i-th feature and defined the Shapley value A_i^* as follows:

$$A_i^* = \sum_{P \subset \Omega \setminus \{i\}} \frac{|P|!(|\Omega| - |P| - 1)!}{|P|!} \left[F(I_{P \cup \{i\}}) - F(I_P) \right] \tag{1}$$

It is the unique solution that satisfies desirable properties to assign attribution value to each feature dimension in the input (Chen et al., 2018). Appendix A shows properties of the Shapley value.

3.2 Evaluating the bias of the attribution map at the pixel level

In this section, we design a metric to accurately evaluate the objectiveness of the attribution map. Given an image $I \in \mathbf{I}$, let us consider the DNN F with a single scalar output $y = F(I)$. For DNNs with multiple outputs, existing methods usually explain each individual output dimension independently. Let $\{a_i\}$ denote the pixel-wise attribution map estimated by a specific explanation method. We aim to evaluate the bias of $\{a_i\}$. People usually formulate the network output as the sum of pixel-wise attribution value, *i.e.* the output y can be decomposed as follows.

$$y = b + \sum_{i \in \Omega} A_i, \qquad \text{s.t.} \quad A_i = \lambda a_i \tag{2}$$

b denotes the bias; i denotes the index of each pixel in the input image; Ω denotes the set of all pixels in the image. Aforementioned $\{A_i^*\}$ can be considered as the ground-truth of $\{A_i\}$ (Lundberg & Lee, 2017). Since many explanation methods (Selvaraju et al., 2017; Simonyan et al., 2013) mainly compute relative values of attributions $\{a_i\}$, instead of a strict attribution map $\{A_i\}$. We use λ to bridge $\{A_i\}$ and $\{a_i\}$. λ is a constant for normalization, which can be eliminated during the implementation of the evaluation.

The estimated attribution of each pixel can be assumed to follow a Gaussian distribution $A_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ (Castro et al., 2009). Attribution distributions of different pixels can be further assumed to share a unified variance, *i.e.* $\sigma_1^2 \approx \sigma_2^2 \approx ... \approx \sigma_n^2$. The evaluation of the attribution distribution $\{a_i\}$ has two aspects, *i.e.*
1. the sampling of pixels whose attributions that are more likely to have large deviations;
2. the evaluation of the bias of the sampled attributions.

First, for the sampling of attributions of interest, we sample the set of pixels S with top-ranked high (or low) attributions. Attribution values of pixels in S are sampled as those with the highest (or the lowest) values, and these pixels are supposed to be more likely to be significantly biased towards high (or low) attribution values. Meanwhile, from another perspective, the distribution of the sampled attribution values is close to the Gumbel distribution.

Second, although the Shapley value can be considered as a standard formulation of the pixel-wise attribution, it usually cannot be accurately computed because of its high computational cost. In order to accurately evaluate the sampled attribution values without significantly increasing the computational cost, we applied the Shapley value approximated by the sampling method. Just like the target attribution distribution A_i, the approximated Shapley value A_i^{shap} is assumed to follow a normal distribution $\mathcal{N}(A_i^*, (\sigma^{shap})^2)$. A_i^{shap} is an unbiased approximation of the true Shapley value A_i^*, Thus, the average value over different pixels in S satisfies $\frac{\sum_{i \in S} A_i^{shap}}{|S|} \sim \mathcal{N}(\frac{\sum_{i \in S} A_i^*}{|S|}, \frac{(\sigma^{shap})^2}{|S|})$.

We can prove that the measurement of the average attribution among all sampled pixels $\frac{\sum_{i \in S} A_i^{shap}}{|S|}$ is of much higher accuracy than the raw Shapley value with the same computational cost. The difference between the highest (or lowest) values and its true values $\left| \frac{\sum_{i \in S} A_i^{shap}}{|S| \| A^{shap} \|} - \frac{\sum_{i \in S} A_i}{|S| \| A \|} \right|$ can reflect the system bias, as follows.

$$M_{\text{pixel}} = \mathbb{E}_I \left[\left| \frac{\sum_{i \in S} A_i^{shap}}{|S| \| A^{shap} \|} - \frac{\sum_{i \in S} A_i}{|S| \| A \|} \right| \right] = \mathbb{E}_I \left[\frac{1}{|S|} \left| \frac{\sum_{i \in S} A_i^{shap}}{\| A^{shap} \|} - \frac{\sum_{i \in S} a_i}{\| a \|} \right| \right] \tag{3}$$
$$\text{s.t.} \qquad \forall i \in S, j \in \Omega \setminus S, a_i \geq a_j \quad \text{or} \quad \forall i \in S, j \in \Omega \setminus S, a_i \leq a_j$$

where Ω is the set of all pixels in an image. $\| A^{shap} \|$ and $\| a \|$ are used for normalization. A small value of M_{pixel} indicates the low bias of the attribution map.

Analysis of the high computational efficiency: Suppose that the computational complexity of processing one sample is $\mathcal{O}(N)$, then the computational complexity of sampling m times is $\mathcal{O}(mN)$.

If the raw Shapley value needs to obtain the same accuracy, it needs significantly more samples, and the computational complexity is $\mathcal{O}(|S|mN)$. Please see Appendix B for the theoretical analysis of the save of computational cost.

In addition, the proposed metric can also be used to evaluate the attribution of neural activations in the intermediate layer, such as those generated by Grad-CAM. In this case, we can regard the target intermediate-layer feature as the input image to compute attributions, so as to implement the evaluation. For each image, we need to sample multiple times to increase the accuracy. We compute the average performance over different images for evaluation. We need to sample multiple times with different images to increase the accuracy of the evaluation. Note that although the metric is designed based on the Shapley value, experimental results showed that LRP outperforms DeepSHAP.

3.3 QUANTIFICATION OF UNEXPLAINABLE FEATURE COMPONENTS

We propose another metric to quantify unexplainable feature components. Given an image I and its attribution map $\{a_i\}$, we generate a new image, which reflects the logic of the attribution map. In this way, we can consider the feature of the newly generated image $\widetilde{f}$ as feature components that can be explained. Let f denote the feature of the original image I. Then, $f - \widetilde{f}$ corresponds to the unexplainable feature components.

To generate the new image, we mask specific pixels in the original image I, which have the lowest attributions. We select and mask a set of pixels S with the lowest absolute attributions, and the number of the selected points is determined subjects to $\sum_{i\in S}|a_i| = 0.1\sum_{i\in\Omega}|a_i|$ to generate the new image $\widetilde{I}$. The metric is formulated as

$$M_{\text{feature}} = \alpha\mathbb{E}_I\left[\|\widetilde{f}_I - f_I\|\right] \tag{4}$$

where $\alpha = \frac{1}{\mathbb{E}_{I'}[\|f_{I'} - \mathbb{E}_{I''}[f_{I''}]\|]}$ is used for normalization. A small value of M_{feature} indicates most feature components in f are explainable.

3.4 EVALUATING THE ROBUSTNESS OF EXPLANATION

This metric is used to measure the robustness of explanation methods to the spatial masking. We believe that the method, which is robust to spatial masking, can be considered more convincing. The robustness is an important perspective of evaluating explanation methods.

Given an input image $I \in \mathbf{I}$ and its attribution map $\{a_i\}$ *w.r.t* a DNN, we use a mask M to cover specific parts of the image to get a masked image $\hat{I}$. For each input image I, we can generate four masked images by masking the right, left, top, and bottom half of the image, respectively. For each masked image $\hat{I}$, the explanation method estimates the attribution $\hat{a}_i$ for each pixel. We compare pixel-level attributions of the unmask pixels between original images and masked images, as follows.

$$M_{\text{non-robust}} = \mathbb{E}_I\left[\frac{1}{\|a\|}\sqrt{\sum_{i\in I\setminus I_{mask}}(a_i - \hat{a}_i)^2}\right] \tag{5}$$

We used $\|a\|$ for normalization, and a large value of $M_{\text{non-robust}}$ indicates a high non-robustness.

3.5 EVALUATING THE MUTUAL VERIFICATION

This metric aims to quantitatively measure the mutual verification between different explanation methods. It is usually believed that two explanation methods are more reliable if they can verify each other. Given a DNN F and an image $I \in \mathbf{I}$, two different explanation methods α and β produce attribution maps a_α and a_β, respectively. We measure their difference as follows.

$$M_{\text{mutual}} = \mathbb{E}_I\left[\|\frac{a_\alpha}{\|a_\alpha\|} - \frac{a_\beta}{\|a_\beta\|}\|\right] \tag{6}$$

Attribution maps from different explanation methods are normalized by their L2-norm. A lower value of M_{mutual} indicates a more convincing mutual verification between explanation methods α and β.

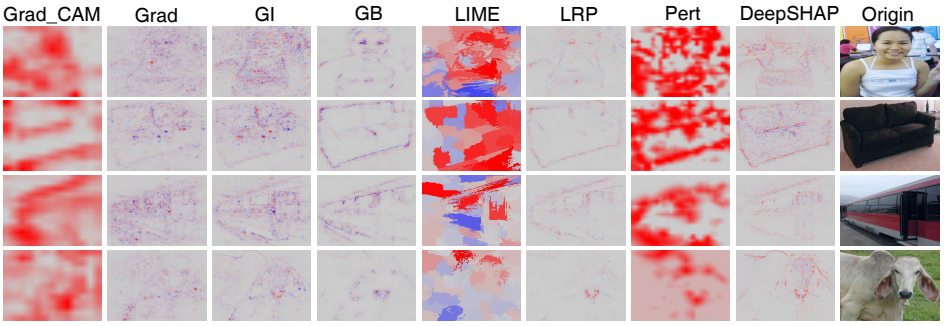

Figure 1: Example of attribution maps of different methods.

4 EXPERIMENT

To evaluate explanation methods, we conducted experiments on the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset and the Pascal VOC 2012 (Everingham et al., 2010) dataset. For images in the Pascal VOC 2012 dataset, we cropped objects using their bounding boxes and used the cropped objects as inputs to train DNNs for object classification. We trained and explained LeNet (LeCun et al., 1998), ResNet-20/32/44/56 (He et al., 2016) using the CIFAR-10 dataset. AlexNet (Krizhevsky et al., 2012), VGG-16/19 (Simonyan & Zisserman, 2015), ResNet-50/101 (He et al., 2016) were trained using the Pascal VOC 2012 dataset.

4.1 BASELINE

In our experiments, we mainly evaluated the following explanation methods. Figure 1 shows attribution maps yielded by these explanation methods. Appendix D provides more attribution maps.

Grad: Given an input, (Simonyan et al., 2013) quantified the attribution value with the gradient of the input. We termed this algorithm as Grad. For RGB images with multiple channels, Grad selected the maximum magnitude across all channels for each pixel.
GI: (Shrikumar et al., 2016) proposed a method, namely GI, which used the pixel-wise product of the input and its gradient as attribution value. Attribution values for RGB channels were summed to get the final attribution value.
GB: Guided Back-propagation, namely GB, corresponded to Grad where the back-propagation rule at ReLU units was redefined (Springenberg et al., 2014).
LRP: Layer-wise relevance propagation (LRP) (Binder et al., 2016) redefined back-propagation rules for each layer to decompose the output of a DNN over the input. We used LRP-ϵ and set the parameter $\epsilon = 1$ in experiments.
DeepSHAP: DeepSHAP adapted DeepLIFT (Shrikumar et al., 2016) to approximate pixel-wise Shapley values for the input image (Lundberg & Lee, 2017). We applied the code released by (Lundberg & Lee, 2017).
LIME: LIME (Ribeiro et al., 2016) trained an interpretable model to compute the attribute value for each super-pixel. We used the code released by (Ribeiro et al., 2016).
Pert: (Fong & Vedaldi, 2017) explained a prediction by training a mask to perturb the input image. Mask values ranging between 0 and 1 indicated the saliency of each pixel. We termed this method Pert.
CAM: CAM computed attribution map over the feature from the last convolutional layer (Zhou et al., 2016). It required the special structure with a global average pooling layer and a fully connected layer at the end of the DNN.
Grad_CAM: Grad_CAM was similar to CAM (Selvaraju et al., 2017). Grad_CAM used gradients over the feature map, instead of the parameters of the fully connected layer.

4.2 IMPLEMENTATION DETAILS

Bias of the attribution map at the pixel level: To approximate the Shapley value for each image, we sampled 1000 times for each image in the CIFAR-10 dataset and sampled 100 times for each image in the Pascal VOC 2012 dataset. We sampled the top-10%, 30%, 50%, 70%, 90% pixels with the highest/lowest values.

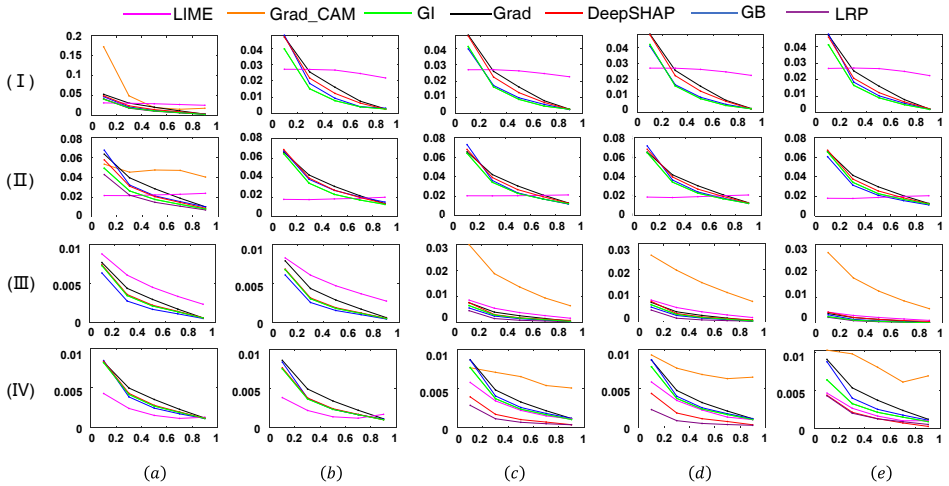

Figure 2: Results of the bias of the attribution map at the pixel level. Row (I) and row (II) used trained LeNet, ResNet20, ResNet32, ResNet44, ResNet56 on the CIFAR-10 dataset from left to right; row (III) and row (IV) used trained ResNet50, ResNet101, VGG16, VGG19, AlexNet on the Pascal VOC 2012 dataset from left to right. Row (I) and row (III) sampled pixels with the highest attribution values; row (II) and row (IV) sampled the pixels with the lowest attribution values.

Table 2: Quantification of unexplainable feature components.

Method	Grad	GI	GB	DeepSHAP	LIME	LRP	Pert
CIFAR10-LeNet	0.96821	1.10399	0.91546	1.10371	**0.67032**	1.08469	0.70177
CIFAR10-ResNet20	1.22212	1.28065	1.14596	1.26002	**1.05286**	-	1.13032
CIFAR10-ResNet32	1.22803	1.29324	1.16028	1.27681	**1.04412**	-	1.14261
CIFAR10-ResNet44	1.22382	1.26434	1.10411	1.24961	**1.02039**	-	1.10772
CIFAR10-ResNet56	1.22487	1.24123	1.11306	1.24146	**1.00916**	-	1.10122
VOC2012-AlexNet	1.02425	1.09375	1.04981	1.0809	**1.01942**	1.11513	1.04048
VOC2012-VGG16	1.18959	1.22715	1.22724	1.32364	**1.12339**	1.32674	1.12878
VOC2012-VGG19	**1.08084**	1.1226	1.10411	1.17121	1.08387	1.23784	1.13043
VOC2012-ResNet50	1.23345	1.24123	**1.2156**	1.23441	1.22411	-	1.25732
VOC2012-ResNet101	1.08592	1.09363	**1.07853**	1.07856	1.11289	-	1.23279

Quantification of unexplainable feature components: Given an image, we masked the pixels with the lowest absolute attribution value. The number of the masked pixels was determined to ensure that the sum of masked absolute attribution value took 10% of the total absolute attribution value. On average, around 30% pixels were masked. The masked pixels were assigned with the average pixel value over images. We used features of the last convolutional layer to compute $M_{feature}$.

4.3 Experiment Result and Analysis

Bias of the attribution map at the pixel level: Figure 2 shows curves of evaluation results on different models learned using different datasets. According to these curves, GI and LIME provided the least biased attribution maps for ResNet at the pixel level. For AlexNet, VGG-16/19 and LeNet, LRP outperformed other methods. Detailed numbers corresponding curves in Figure 2 are listed in the Appendix E.

Some methods could not be evaluated using the bias at the pixel level. For example, Pert computed an importance map without negative values instead of an attribution map for each image. The code of CAM (Zhou et al., 2016) projected attribution values to the range between 0 and 1. Grad_CAM and LRP were not used on residual networks, because there was only one fully connected layer behind the last convolutional layer in residual networks. In this case, Grad_CAM could not diagnose the logic contained in the cascaded non-liner layers of the DNN. For LRP, the relevance propagation rules of some structures in ResNet were not defined to the best of our knowledge.

Quantification of unexplainable feature components: Table 2 compares the amount of unexplainable feature components between explanation methods. We found that LIME, GB and Pert explained more feature components than other methods. We noticed that the quantification of unexplainable feature components of most explanation methods were considerable larger than expected. It was be-

Table 3: Non-robustness of explanation methods with different datasets/models.

Method	Grad	GI	GB	DeepSHAP	LIME	LRP	Pert	CAM	Grad_CAM
CIFAR10-LeNet	0.510	0.610	0.548	0.843	-	0.932	**0.369**	-	1.039
CIFAR10-ResNet20	0.515	0.599	**0.330**	0.541	-	-	0.466	0.443	0.386
CIFAR10-ResNet32	0.762	0.832	0.321	0.645	-	-	0.469	0.351	**0.318**
CIFAR10-ResNet44	0.957	1.010	0.299	0.513	-	-	0.470	0.321	**0.287**
CIFAR10-ResNet56	0.911	0.959	**0.298**	0.567	-	-	0.475	0.341	0.321
VOC2012-AlexNet	0.519	0.603	**0.421**	0.448	3.994	0.719	0.602	-	0.489
VOC2012-VGG16	0.475	0.500	**0.271**	0.388	2.175	0.447	0.635	-	0.381
VOC2012-VGG19	0.505	0.524	**0.286**	0.356	2.172	0.444	0.666	-	0.381
VOC2012-ResNet50	0.605	0.663	**0.280**	0.936	2.354	-	0.552	0.367	0.309
VOC2012-ResNet101	0.553	0.593	0.269	0.847	4.667	-	0.565	0.347	**0.266**

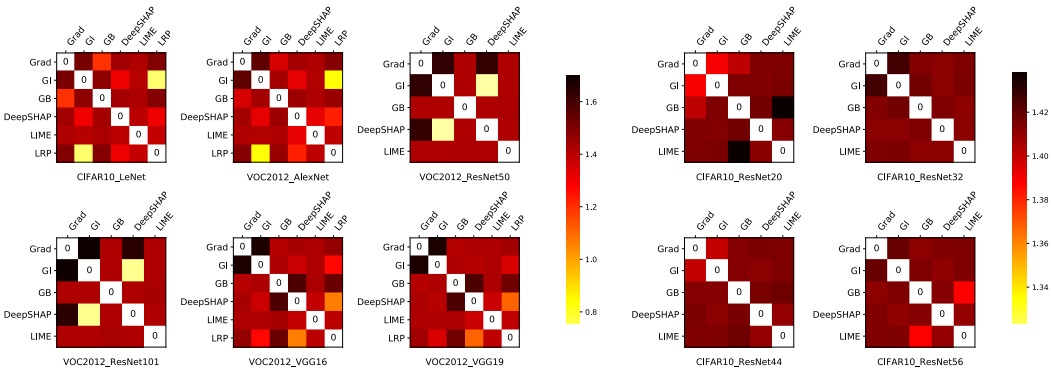

Figure 3: Heat maps of mutual verification. A low value of M_{mutual} between two methods indicates a more convincing mutual verification between them.

cause the attribution maps from some methods contained relatively larger noise. Thus, the masked pixels were almost uniformly distributed over images, which destroyed the context information and led to worse results.

We did not evaluate CAM and Grad_CAM, because they calculated attribution maps at the feature level, which were not comparable with attribution values at the pixel level.

Robustness of explanation: Table 3 shows the quantitative results of $M_{\text{non-robust}}$ on different models trained using different datasets. We found that GB and Grad_CAM exhibited a lower non-robustness to spatial masking. For more results, Appendix F shows examples of attribution maps of masked images.

Mutual verification: Figure 3 visualizes the mutual verification M_{mutual} between different explanation methods, which indicates a high level mutual verification between LRP, GI and DeepSHAP. Appendix G provides more detailed results. Note that we did not compare CAM and Grad_CAM with other methods. It was because they computed attribution maps on intermediate-layer features.

5 Conclusion

In this paper, we have proposed four metrics to evaluate explanation methods from four different perspectives. The proposed evaluation metrics are computed without requirements for ground-truth explanations. Our metrics can be broadly applied to different methods, *w.r.t.* DNNs learned using different datasets. These metrics evaluate the bias of the attribution map at the pixel level, quantify the unexplainable feature components, the robustness of the explanation and the mutual verification. In experiments, we used our metrics to evaluate nine widely used explanation methods. Experimental results showed that attribution maps from LRP, GI and LIME exhibited lower bias at the pixel level. LIME and GB explained more feature components than other methods. Regarding the robustness, GB, CAM and Grad_CAM were more robust to spatial masking than other explanation methods. DeepSHAP, GI and LRP can better verified each other.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *In NIPS*, 2018.

David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. *In arXiv:1806.07538*, 2018.

Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. Evaluating recurrent neural network explanations. *In arXiv:1904.11829*, 2019.

Mathieu Aubry and Bryan C. Russell. Understanding deep features with computer-generated imagery. *In ICCV*, 2015.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. *In CVPR*, 2017.

Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. *In International Conference on Artificial Neural Networks (ICANN)*, 2016.

Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *In Computers & Operations Research*, 36(5):1726–1730, 2009.

Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. Interpretable deep models for icu outcome prediction. *In American Medical Informatics Association (AMIA) Annual Symposium*, 2016.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. *In arXiv:1808.02610*, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *In NIPS*, 2016.

Edward Choi, Mohammad Taha Bahadori, Joshua A. Kulas, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Retain: an interpretable predictive model for healthcare using reverse time attention mechansim. *In arXiv:1608.05745v4*, 2017.

Xiaocong Cui, Jung Min Lee, and J Hsieh. An integrative 3c evaluation framework for explainable artificial intelligence. *In The annual Americas Conference on Information Systems (AMCIS)*, 2019.

Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. *In CVPR*, 2016.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *In International Journal of Computer Vision*, 88(2):303–338, June 2010.

Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. *In CVPR*, 2018.

Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *In arXiv:1704.03296v1*, 2017.

Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *In arXiv:1711.09784*, 2017.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *In AAAI*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *In CVPR*, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-vae: learning basic visual concepts with a constrained variational framework. *In ICLR*, 2017.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric P. Xing. Harnessing deep neural networks with logic rules. *In arXiv:1603.06318v2*, 2016.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *In arXiv:1711.11279*, 2017.

Pieter-Jan Kindermans, Kristof T. Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. Learning how to explain neural networks: Patternnet and patternattribution. *In ICLR*, 2018.

PangWei Koh and Percy Liang. Understanding black-box predictions via influence functions. *In ICML*, 2017.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *In Computer Science Department, University of Toronto, Tech. Rep*, 1, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *In NIPS*, 2012.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. *In AAAI*, 2017.

Yann LeCun, Lèon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *In Proceedings of the IEEE*, 1998.

Renjie Liao, Alex Schwing, Richard Zemel, and Raquel Urtasun. Learning deep parsimonious representations. *In NIPS*, 2016.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *In NIPS*, 2017.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. *In CVPR*, 2015.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. *In KDD*, 2016.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. *In NIPS*, 2017.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *In ICCV*, 2017.

Lloyd S Shapley. A value for n-person games. *In Contributions to the Theory of Games*, 2(28): 307–317, 1953.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *In arXiv:1605.01713*, 2016.

Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. *In ICCV*, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *In ICLR*, 2015.

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

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

Austin Stone, Huayan Wang, Yi Liu, D. Scott Phoenix, and Dileep George. Teaching compositionality to cnns. *In CVPR*, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *In arXiv:1312.6199v4*, 2014.

Sarah Tan, Rich Caruana, Giles Hooker, and Albert Gordo. Transparent model distillation. *In arXiv:1801.08640*, 2018.

Joel Vaughan, Agus Sudjianto, Erind Brahimi, Jie Chen, and Vijayan N. Nair. Explainable neural networks based on additive index models. *In arXiv:1806.01933*, 2018.

Minh N Vu, Truc D Nguyen, NhatHai Phan, Ralucca Gera, and My T Thai. Evaluating explainers via perturbation. *In arXiv:1906.02032*, 2019.

Mike Wu, Michael C. Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. *In NIPS TIML Workshop*, 2017a.

Tianfu Wu, Xilai Li, Xi Song, Wei Sun, Liang Dong, and Bo Li. Interpretable r-cnn. *In arXiv:1711.05226*, 2017b.

Mengjiao Yang and Been Kim. Bim: Towards quantitative evaluation of interpretability methods with ground truth. *In arXiv:1907.09701*, 2019.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *In NIPS*, 2014.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *In ICML Deep Learning Workshop*, 2015.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *In ECCV*, 2014.

Quanshi Zhang, Yu Yang, Yuchen Liu, Ying Nian Wu, and Song-Chun Zhu. Unsupervised learning of neural networks to explain neural networks. *In arXiv:1805.07468*, 2018.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *In ICRL*, 2015.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *In CVPR*, 2016.

A PROPERTIES OF THE SHAPLEY VALUE

Let I denote the input image; let Ω denote the set of all pixels in I. We can use $I_\emptyset$ to denote a baseline image, *i.e.* all pixels in $I_\emptyset$ equal to the average value over all images. For a subset $S \subset \Omega$, I_S denotes an image that satisfies

$$(I_S)_i = \begin{cases} (I)_i, & i \in S \\ (I_\emptyset)_i, & i \notin S \end{cases} \tag{7}$$

where i is the index of the pixel in I and I_Ω is the same image as I. Let F and G denote two models with scalar output. The Shapley value of the i-th pixel is represented by A_i^*, and they have the following properties.

Efficiency: The sum of Shapley values $\sum_{i\in\Omega} A_i^*$ is equal to $F(I_\Omega) - F(I_\emptyset)$.

Symmetry: The features that are treated equally by the model are treated equally by the Shapley value. If $F(I_{S\cup\{i\}}) = F(I_{S\cup\{j\}})$ for all subsets S, then $A_i^* = A_j^*$.

Additivity: For any two models F and G, if they are combined into one model $F + G$, the Shapley value must be added pixel by pixel: $(A^*)_i^{F+G} = (A^*)_i^F + (A^*)_i^G$.

Monotonicity: For any two models F and G, if for all subsets S we have $F(I_{S\cup\{i\}}) - F(I_S) \geq G(I_{S\cup\{i\}}) - G(I_S)$ for all subsets S, then we have $(A^*)_i^F \geq (A^*)_i^G$.

B ANALYSIS OF THE COMPUTATIONAL COST

In this section, we continue using the notation in Section 3.1 and Section 3.2. Suppose that we sample m times to approximate the Shapley value. The variance of A_i^{shap} is $\frac{\sigma^2}{m}$ where σ^2 satisfies (Castro et al., 2009)

$$\sigma^2 = \sum_{P\subset\Omega\setminus\{i\}} \frac{|P|!(|\Omega| - |P| - 1)!}{|P|!} \left[F(I_{P\cup\{i\}}) - F(I_P) - A_i^*\right]^2 \tag{8}$$

So we have $(\sigma^{shap})^2 = \sigma^2/m$. For the set of sampled pixels S, the variance of their average Shapley value is $\frac{|S|(\sigma^{shap})^2}{|S|^2} = \frac{(\sigma^{shap})^2}{|S|} = \frac{\sigma^2}{m|S|}$. Apparently, if we want to get the same accuracy for a single pixel as the set of pixels, we need to sample $m|S|$ times, which needs much more computational cost than our metric, especially when the number of sampled pixels is large.

C STUDIES OF EXPLANATION METHODS BESIDES THE ESTIMATION OF ATTRIBUTION MAPS.

Network visualization: The visualization of feature representations inside a neural network is the most direct way of opening the black-box of the neural network. Related techniques include gradient-based visualization (Zeiler & Fergus, 2014; Mahendran & Vedaldi, 2015; Yosinski et al., 2015) and up-convolutional nets (Dosovitskiy & Brox, 2016) to invert feature maps of conv-layers into images.

Network diagnosis: Some studies diagnose feature representations inside a neural network. (Yosinski et al., 2014) measured features transferability in intermediate layers of a neural network. (Aubry & Russell, 2015) visualized feature distributions of different categories in the feature space. (Kindermans et al., 2018) extracted rough pixel-level correlations between network inputs and outputs, *i.e.* estimating image regions that directly contribute the network output. Network-attack methods (Koh & Liang, 2017; Szegedy et al., 2014) computed adversarial samples to diagnose a CNN. (Lakkaraju et al., 2017) discovered knowledge blind spots of a CNN in a weakly-supervised manner. However, above methods usually analyzed a neural network at the pixel level and did not summarize the network knowledge into clear visual concepts. (Bau et al., 2017) defined six types of semantics for CNN filters, *i.e.* objects, parts, scenes, textures, materials, and colors. Then, (Zhou et al., 2015) proposed a method to compute the image-resolution receptive field of neural activations in a feature map. Fong and Vedaldi (Fong & Vedaldi, 2018) analyzed how multiple filters jointly represented

a certain semantic concept. Other studies retrieved intermediate-layer features from CNNs representing clear concepts. Simon & Rodner (2015) retrieved features to describe objects from feature maps, respectively. (Zhou et al., 2015) selected neural units to describe scenes.

Learning interpretable representations: A new trend in the scope of network interpretability is to learn interpretable feature representations in neural networks Hu et al. (2016); Stone et al. (2017); Liao et al. (2016) in an un-/weakly-supervised manner. Capsule nets Sabour et al. (2017) and interpretable RCNN Wu et al. (2017b) learned interpretable features in intermediate layers. InfoGAN Chen et al. (2016) and β-VAE Higgins et al. (2017) learned well-disentangled codes for generative networks.

Explaining neural networks via knowledge distillation: Distilling knowledge from a black-box model into an explainable model is an emerging direction in recent years. (Choi et al., 2017) learned an explainable additive model, and (Vaughan et al., 2018) distilled knowledge of a network into an additive model. In order to disentangle feature representations of object parts from intermediate layers of a CNN, (Zhang et al., 2018) distilled the CNN's knowledge into an explainer network with interpretable conv-layers, in which each filter represented a specific object part. (Frosst & Hinton, 2017; Tan et al., 2018; Che et al., 2016; Wu et al., 2017a) distilled representations of neural networks into tree structures. These methods did not explain the network knowledge using human-interpretable semantic concepts.

D MORE EXAMPLES OF ATTRIBUTION MAPS

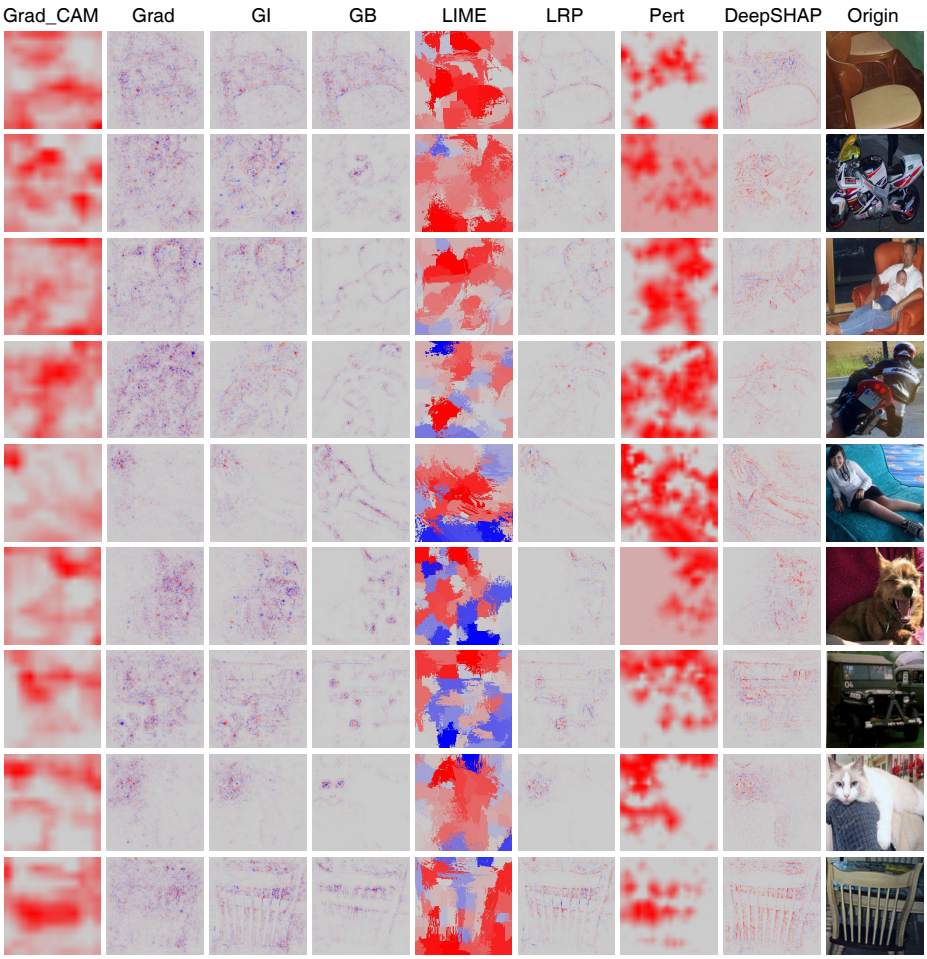

Figure 4: Example of attribution maps.

E DETAILED RESULTS OF THE BIAS OF THE ATTRIBUTION MAP AT THE PIXEL LEVEL

Table 4: Bias of the attribution map at the pixel level on CIFAR-10-LeNet

Method	Grad_CAM	Grad	GI	GB	DeepSHAP	LIME	LRP
top-10%	0.17256	0.05302	0.04060	0.04701	0.04924	**0.03138**	0.04164
top-30%	0.04921	0.03090	0.01892	0.02109	0.02408	0.03051	**0.01798**
top-50%	0.01961	0.02057	0.01126	0.01225	0.01443	0.02919	**0.01041**
top-70%	0.01531	0.01137	0.00671	0.00746	0.00902	0.02727	**0.00634**
top-90%	0.01785	0.00314	**0.00240**	0.00300	0.00334	0.02537	0.00277
bottom-10%	0.05317	0.06326	0.04907	0.06730	0.05755	**0.02160**	0.04271
bottom-30%	0.04510	0.03948	0.02626	0.03224	0.03100	**0.02150**	0.02198
bottom-50%	0.04732	0.02840	0.01767	0.02126	0.02047	0.02222	**0.01481**
bottom-70%	0.04693	0.01886	0.01270	0.01551	0.01462	0.02286	**0.01086**
bottom-90%	0.04022	0.01026	0.00808	0.01025	0.00882	0.02385	**0.00708**

Table 5: Bias of the attribution map at the pixel level on CIFAR-10-ResNet20

Method	Grad	GI	GB	DeepSHAP	LIME
top-10%	0.04692	0.03962	0.04801	0.04702	**0.02716**
top-30%	0.02552	**0.01522**	0.01848	0.02175	0.02706
top-50%	0.01616	**0.00795**	0.00943	0.01241	0.02666
top-70%	0.00762	**0.00416**	0.00425	0.00643	0.02456
top-90%	0.00288	0.00283	0.00336	**0.00270**	0.02185
bottom-10%	0.06610	0.06449	0.06726	0.06866	**0.01790**
bottom-30%	0.04272	0.03437	0.03833	0.03962	**0.01765**
bottom-50%	0.03117	0.02296	0.02704	0.02722	**0.01836**
bottom-70%	0.02168	0.01726	0.02053	0.01992	**0.01897**
bottom-90%	0.01357	**0.01275**	0.01513	0.01346	0.02005

Table 6: Bias of the attribution map at the pixel level on CIFAR-10-ResNet32

Method	Grad	GI	GB	DeepSHAP	LIME
top-10%	0.04836	0.04126	0.03984	0.04773	**0.02699**
top-30%	0.02600	**0.01641**	0.01728	0.02247	0.02702
top-50%	0.01639	**0.00874**	0.00972	0.01289	0.02630
top-70%	0.00793	**0.00459**	0.00565	0.00683	0.02465
top-90%	0.00268	0.00264	**0.00242**	0.00260	0.02267
bottom-10%	0.06535	0.06389	0.07271	0.06787	**0.02059**
bottom-30%	0.04217	0.03433	0.03612	0.03917	**0.02053**
bottom-50%	0.03064	0.02304	0.02359	0.02686	**0.02065**
bottom-70%	0.02135	**0.01726**	0.01733	0.01963	0.02103
bottom-90%	0.01331	0.01254	**0.01214**	0.01307	0.02145

Table 7: Bias of the attribution map at the pixel level on CIFAR-10-ResNet44

Method	Grad	GI	GB	DeepSHAP	LIME
top-10%	0.04795	0.04155	0.04055	0.04751	**0.02712**
top-30%	0.02576	**0.01626**	0.01698	0.02254	0.02709
top-50%	0.01609	**0.00848**	0.00935	0.01307	0.02637
top-70%	0.00748	**0.00433**	0.00502	0.00686	0.02485
top-90%	0.00248	0.00228	0.00230	**0.00226**	0.02271
bottom-10%	0.06510	0.06452	0.07143	0.06802	**0.01924**
bottom-30%	0.04196	0.03449	0.03663	0.03977	**0.01879**
bottom-50%	0.03082	0.02323	0.02448	0.02736	**0.01967**
bottom-70%	0.02159	**0.01752**	0.01811	0.01989	0.02064
bottom-90%	0.01353	**0.01282**	0.01293	0.01323	0.02137

Table 8: Bias of the attribution map at the pixel level on CIFAR-10-ResNet56

Method	Grad	GI	GB	DeepSHAP	LIME
top-10%	0.04716	0.04129	0.04780	0.04601	**0.02690**
top-30%	0.02537	**0.01659**	0.01855	0.02089	0.02714
top-50%	0.01607	**0.00900**	0.01022	0.01215	0.02679
top-70%	0.00791	**0.00482**	0.00569	0.00652	0.02509
top-90%	0.00212	**0.00194**	0.00214	0.00234	0.02252
bottom-10%	0.06625	0.06551	0.06065	0.06727	**0.01828**
bottom-30%	0.04164	0.03449	0.03158	0.03779	**0.01812**
bottom-50%	0.03001	0.02299	0.02129	0.02574	**0.01911**
bottom-70%	0.02086	**0.01712**	0.01587	0.01868	0.02023
bottom-90%	0.01300	0.01237	**0.01146**	0.01267	0.02088

Table 9: Bias of the attribution map at the pixel level on VOC2012-ResNet50

Method	Grad	GI	GB	DeepSHAP	LIME
top-10%	0.00766	0.00728	**0.00633**	0.00738	0.00876
top-30%	0.00437	0.00345	**0.00276**	0.00359	0.00606
top-50%	0.00297	0.00213	**0.00170**	0.00223	0.00451
top-70%	0.00173	0.00138	**0.00113**	0.00144	0.00335
top-90%	0.00060	0.00060	**0.00053**	0.00062	0.00235
bottom-10%	0.00831	0.00824	0.00850	0.00839	**0.00430**
bottom-30%	0.00500	0.00417	0.00386	0.00432	**0.00240**
bottom-50%	0.00354	0.00269	0.00244	0.00280	**0.00151**
bottom-70%	0.00228	0.00189	0.00171	0.00194	**0.00113**
bottom-90%	0.00117	0.00112	**0.00111**	0.00114	0.00128

Table 10: Bias of the attribution map at the pixel level on VOC2012-ResNet101

Method	Grad	GI	GB	DeepSHAP	LIME
top-10%	0.00790	0.00678	**0.00610**	0.00685	0.00827
top-30%	0.00438	0.00307	**0.00260**	0.00318	0.00608
top-50%	0.00291	0.00185	**0.00158**	0.00194	0.00473
top-70%	0.00175	0.00121	**0.00104**	0.00127	0.00371
top-90%	0.00067	0.00055	**0.00049**	0.00056	0.00278
bottom-10%	0.00861	0.00754	0.00839	0.00767	**0.00387**
bottom-30%	0.00496	0.00369	0.00377	0.00381	**0.00222**
bottom-50%	0.00343	0.00237	0.00238	0.00245	**0.00142**
bottom-70%	0.00225	0.00169	0.00169	0.00173	**0.00124**
bottom-90%	0.00117	**0.00104**	0.00112	0.00105	0.00174

Table 11: Bias of the attribution map at the pixel level on VOC2012-AlexNet

Method	Grad_CAM	Grad	GI	GB	DeepSHAP	LIME	LRP
top-10%	0.05403	0.00744	0.00505	0.00646	0.00845	0.00854	**0.00453**
top-30%	0.03467	0.00434	0.00243	0.00296	0.00447	0.00598	**0.00197**
top-50%	0.02459	0.00298	0.00149	0.00182	0.00295	0.00443	**0.00119**
top-70%	0.01711	0.00176	0.00094	0.00120	0.00209	0.00329	**0.00080**
top-90%	0.01115	0.00060	**0.00035**	0.00058	0.00140	0.00222	0.00043
bottom-10%	0.01026	0.00906	0.00638	0.00876	0.00438	0.00466	**0.00430**
bottom-30%	0.00978	0.00534	0.00327	0.00400	0.00215	0.00267	**0.00202**
bottom-50%	0.00803	0.00372	0.00215	0.00253	0.00131	0.00165	**0.00129**
bottom-70%	0.00610	0.00239	0.00151	0.00178	**0.00075**	0.00103	0.00091
bottom-90%	0.00690	0.00124	0.00093	0.00111	**0.00030**	0.00104	0.00056

Table 12: Bias of the attribution map at the pixel level on VOC2012-VGG16

Method	Grad_CAM	Grad	GI	GB	DeepSHAP	LIME	LRP
top-10%	0.02984	0.00764	0.00656	0.00563	0.00768	0.00864	**0.00461**
top-30%	0.01876	0.00405	0.00292	0.00253	0.00310	0.00554	**0.00159**
top-50%	0.01365	0.00267	0.00176	0.00154	0.00187	0.00387	**0.00092**
top-70%	0.00947	0.00162	0.00115	0.00098	0.00132	0.00272	**0.00065**
top-90%	0.00650	0.00061	0.00051	**0.00038**	0.00089	0.00172	0.00044
bottom-10%	0.00767	0.00869	0.00768	0.00865	0.00396	0.00580	**0.00288**
bottom-30%	0.00708	0.00483	0.00372	0.00406	0.00173	0.00344	**0.00117**
bottom-50%	0.00654	0.00330	0.00238	0.00261	0.00106	0.00223	**0.00072**
bottom-70%	0.00540	0.00219	0.00170	0.00185	0.00075	0.00153	**0.00054**
bottom-90%	0.00509	0.00120	0.00107	0.00122	0.00041	0.00114	**0.00040**

Table 13: Bias of the attribution map at the pixel level on VOC2012-VGG19

Method	Grad_CAM	Grad	GI	GB	DeepSHAP	LIME	LRP
top-10%	0.02564	0.00761	0.00672	0.00580	0.00802	0.00851	**0.00468**
top-30%	0.01991	0.00403	0.00301	0.00261	0.00337	0.00565	**0.00157**
top-50%	0.01517	0.00263	0.00181	0.00159	0.00205	0.00402	**0.00091**
top-70%	0.01139	0.00161	0.00118	0.00102	0.00143	0.00282	**0.00064**
top-90%	0.00799	0.00062	0.00053	**0.00043**	0.00096	0.00183	0.00045
bottom-10%	0.00932	0.00867	0.00782	0.00871	0.00442	0.00587	**0.00236**
bottom-30%	0.00764	0.00480	0.00378	0.00406	0.00193	0.00352	**0.00095**
bottom-50%	0.00683	0.00326	0.00243	0.00260	0.00120	0.00231	**0.00059**
bottom-70%	0.00630	0.00217	0.00173	0.00184	0.00084	0.00140	**0.00046**
bottom-90%	0.00649	0.00119	0.00109	0.00120	0.00043	0.00107	**0.00036**

F ATTRIBUTION MAPS OF MASKED IMAGES

G DETAILED RESULTS OF THE MUTUAL VERIFICATION

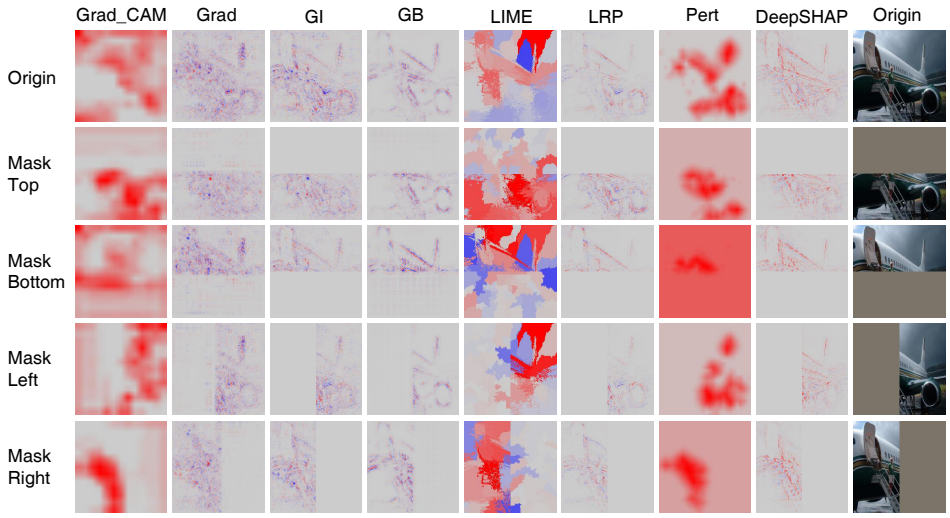

Figure 5: Example of attribution maps after spatial masking.

Table 14: Mutual verification on VOC2012-VGG19

Method	Grad	GI	GB	DeepSHAP	LIME	LRP
Grad	0.0000	1.6661	1.4054	1.4177	1.4142	1.4375
GI	1.6661	0.0000	1.4159	1.4008	1.4140	1.3437
GB	1.4054	1.4159	0.0000	1.5834	1.4244	1.5242
DeepSHAP	1.4177	1.4009	1.5835	0.0000	1.3691	**1.1106**
LIME	1.4142	1.4141	1.4245	1.3691	0.0000	1.3874
LRP	1.4375	1.3437	1.5243	**1.1106**	1.3874	0.0000

Table 15: Mutual verification on VOC2012-VGG16

Method	Grad	GI	GB	DeepSHAP	LIME	LRP
Grad	0.0000	1.6647	1.3992	1.4302	1.4142	1.4621
GI	1.6647	0.0000	1.4191	1.3711	1.4139	1.2649
GB	1.3992	1.4191	0.0000	1.5839	1.4260	1.5320
DeepSHAP	1.4302	1.3712	1.5840	0.0000	1.3824	**1.0636**
LIME	1.4142	1.4140	1.4261	1.3825	0.0000	1.3960
LRP	1.4621	1.2649	1.5321	**1.0636**	1.3960	0.0000

Table 16: Mutual verification on VOC2012-AlexNet

Method	Grad	GI	GB	DeepSHAP	LIME	LRP
Grad	0.0000	1.5540	1.3477	1.4348	1.4148	1.4851
GI	1.5541	0.0000	1.4344	1.3224	1.4113	**0.8341**
GB	1.3477	1.4344	0.0000	1.4477	1.4146	1.4573
DeepSHAP	1.4349	1.3225	1.4478	0.0000	1.3221	1.2251
LIME	1.4149	1.4115	1.4147	1.3222	0.0000	1.3916
LRP	1.4851	**0.8341**	1.4573	1.2250	1.3915	0.0000

Table 17: Mutual verification on VOC2012-ResNet50

Method	Grad	GI	GB	DeepSHAP	LIME
Grad	0.0000	1.6285	1.4143	1.6242	1.4143
GI	1.6285	0.0000	1.4137	**0.6537**	1.4139
GB	1.4143	1.4136	0.0000	1.4137	1.4238
DeepSHAP	1.6242	**0.6537**	1.4137	0.0000	1.4139
LIME	1.4144	1.4141	1.4240	1.4140	0.0000

Table 18: Mutual verification on VOC2012-ResNet101

Method	Grad	GI	GB	DeepSHAP	LIME
Grad	0.0000	1.6877	1.4136	1.6428	1.4145
GI	1.6877	0.0000	1.4140	**0.6778**	1.4139
GB	1.4136	1.4140	0.0000	1.4140	1.4275
DeepSHAP	1.6428	**0.6778**	1.4140	0.0000	1.4138
LIME	1.4146	1.4140	1.4277	1.4139	0.0000

Table 19: Mutual verification on CIFAR10-ResNet56

Method	Grad	GI	GB	DeepSHAP	LIME
Grad	0.0000	1.4183	1.4107	1.4135	1.4142
GI	1.4183	0.0000	1.4140	1.4104	1.4140
GB	1.4107	1.4140	0.0000	1.4128	1.3879
DeepSHAP	1.4135	1.4104	1.4128	0.0000	1.4097
LIME	1.4142	1.4140	1.3879	1.4098	0.0000

Table 20: Mutual verification on CIFAR10-ResNet44

Method	Grad	GI	GB	DeepSHAP	LIME
Grad	0.0000	1.3999	1.4128	1.4147	1.4141
GI	1.3999	0.0000	1.4125	1.4114	1.4143
GB	1.4128	1.4125	0.0000	1.4152	1.4175
DeepSHAP	1.4147	1.4114	1.4152	0.0000	1.4097
LIME	1.4142	1.4144	1.4152	1.4175	0.0000

Table 21: Mutual verification on CIFAR10-ResNet32

Method	Grad	GI	GB	DeepSHAP	LIME
Grad	0.0000	1.4259	1.4129	1.4117	1.4143
GI	1.4259	0.0000	1.4164	1.4113	1.4145
GB	1.4129	1.4164	0.0000	1.4144	1.4111
DeepSHAP	1.4117	1.4113	1.4144	0.0000	1.4117
LIME	1.4144	1.4145	1.4112	1.4117	0.0000

Table 22: Mutual verification on CIFAR10-ResNet20

Method	Grad	GI	GB	DeepSHAP	LIME
Grad	0.0000	1.3895	1.4010	1.4134	1.4143
GI	1.3895	0.0000	1.4140	1.4133	1.4145
GB	1.4010	1.4140	0.0000	1.4164	1.4382
DeepSHAP	1.4134	1.4133	1.4164	0.0000	1.4118
LIME	1.4144	1.4145	1.4382	1.4118	0.0000

Table 23: Mutual verification on CIFAR10-LeNet

Method	Grad	GI	GB	DeepSHAP	LIME	LRP
Grad	0.0000	1.5079	1.1955	1.4352	1.4130	1.4904
GI	1.5079	0.0000	1.4714	1.3025	1.4003	**0.7179**
GB	1.1955	1.4714	0.0000	1.4329	1.4249	1.4881
DeepSHAP	1.4352	1.3025	1.4329	0.0000	1.3958	1.3063
LIME	1.4131	1.4004	1.4249	1.3959	0.0000	1.3805
LRP	1.4904	**0.7179**	1.4881	1.3063	1.3804	0.0000