DISTRIBUTION-GUIDED LOCAL EXPLANATION FOR BLACK-BOX CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing local explanation methods provide an explanation for each decision of black-box classifiers, in the form of relevance scores of features according to their contributions. To obtain satisfying explainability, many methods introduce ad hoc constraints into the classification loss to regularize these relevance scores. However, the large information gap between the classification loss and these constraints increases the difficulty of tuning hyper-parameters. To bridge this gap, in this paper we present a simple but effective mask predictor. Specifically, we model the above constraints with a distribution controller, and integrate it with a neural network to directly guide the distribution of relevance scores. The benefit of this strategy is to facilitate the setting of involved hyper-parameters, and enable discriminative scores over supporting features. The experimental results demonstrate that our method outperforms others in terms of faithfulness and explainability. Meanwhile, it also provides effective saliency maps for explaining each decision. The code is available at https://github.com/iclrlocal/.

1 INTRODUCTION

Deep neural networks (DNNs) have achieved high classification accuracy in a wide range of fields, such as computer vision (He et al., 2016; Simonyan & Zisserman, 2014) and natural language processing (Greff et al., 2016; Mikolov et al., 2010). Despite the superior performance, DNN models lack meaningful explanations on how a specific decision is made, and are often regarded as black-boxes. To address this issue, various global and local explanation methods have been proposed. The former group aims to inspect the structures and the parameters of a complex model (Erhan et al., 2009; Chen et al., 2016). The latter group provides users understandable rationale for a specific decision with relevance scores[1] (Simonyan et al., 2014; Du et al., 2018a).

In this paper, we focus on local explanation as it extracts the intuitive evidence behind the decision of each instance. To obtain the relevance scores for local explanation, gradient-based methods compute the partial derivative of the class probability with respect to an input instance. However, instead of directly pointing out why the target class is derived based on input, it is likely to answer the question (Montavon et al., 2018): What makes this instance more or less similar to the target class? To tackle this limitation, perturbation-based explanation methods are proposed. These methods perturb the input and aim to find the smallest region, which alone allows a confident classification or prevents a confident classification once being removed (Dabkowski & Gal, 2017; Fong & Vedaldi, 2017). By applying various ad hoc constraints, these methods improve explainability and maintains faithfulness [2]. Nevertheless, the large information gap between the classification loss and these constraints in turn increases the difficulty of tuning hyper-parameters.

To bridge this gap, we propose a simple but effective mask predictor. The work is built upon the following observation: *a large portion of contributions to each decision are held by only a small fraction of features* (Fong & Vedaldi, 2017; Chattopadhay et al., 2018; Du et al., 2018a). The proposed predictor consists of a mask generator and a distribution controller. The former takes the

[1] In this paper, relevance scores indicate the contributions of features to a specific decision. A high score implies a higher contribution. Besides, we do not discriminate saliency maps and masks, as both indicate the permutation of relevance scores of each input.

[2] Explainability quantifies how easy it is to understand and reason about the explanation; faithfulness estimates the fidelity between the explanation and the decision behaviour of black boxes.

hidden feature maps in black boxes as inputs, and the latter guides the outputs with expected distributions as relevance scores. In this way, the ad hoc constraints for mask generation introduced in previous work are replaced with the distribution controller. Thus, the process of hyper-parameters tuning is transferred into the task of distribution design. We show that, with an easy setting of the involved hyper-parameters in the controller, it can directly lead the relevance scores to right-skewed distributions with long tails (Clauset et al., 2009). As a result, these scores are more discriminative, since a small portion of long-tail high scores correspond to the supporting features, while the majority features have low scores and are regarded as unimportant. An example is shown in Fig.1(a), where the pixels correspond to the tail are highlighted. Besides, the smoothness of masks can also be achieved by unsampling the outputs of the controller at a coarse scale, after the whole predictor is trained with large-scale images. We further denoise these scores for highlighting supporting features without destroying their ranking. Finally, we introduce two metrics for comprehensively evaluating relevance scores in terms of faithfulness and explainability, respectively.

The main contributions of our work are as follows.

- We develop a trainable mask predictor to simplify the formulations of perturbation-based methods. It integrates a distribution controller with a mask generator, aiming to refine the mask towards the desired score distribution. The predictor is optimized solely under the classification loss without additional constraints, which therefore improves the faithfulness of mimicking target black-box models.
- We provide two practical implementations of the controller. As a result, the predictor can establish the right-skewed distributions for relevance scores by monotonically transforming the output of the mask generator. Besides, we show that the involved hyper-parameters can be easily set before the training stage.
- We introduce two metrics to evaluate the quality of scores in terms of faithfulness and explainability, respectively. Meanwhile, the experiments on real-world datasets demonstrate that our method not only obtains higher quantitative performance for explaining the behavior of black boxes, but also provides discriminative masks for intuitive explanation.

2 Related Work

This section reviews local explanation methods for DNNs, which target to identify the relevance score of each feature towards a specific decision (Montavon et al., 2018).

2.1 Gradient-based Local Explanation

To obtain relevance scores, gradient-based methods compute the partial derivative of the class probability with respect to the input by using back-propagation (Simonyan et al., 2014). In general, these methods are advantageous in their high computational efficiency, i.e., using a few forward and backward iterations is sufficient to generate saliency maps. However, these saliency maps based on the naive gradients are visually noisy and hard to be understand. To address this issue, various methods have been proposed. For example, Smooth Grad (Smilkov et al., 2017) addresses the visual noise by introducing noise to inputs repeatedly. Integrated gradient (Fong & Vedaldi, 2017) estimates the global contribution of each feature rather than the local sensitivity. Guided back-prorogation (Springenberg et al., 2014) modifies the gradients of RELU functions by discarding negative values at the back-propagation process. Besides, recent methods proposed to create saliency maps by combining the gradients with the corresponding features. For example, Grad CAM (Selvaraju et al., 2017) and Grad CAM ++ (Chattopadhay et al., 2018) take advantages of high-level feature maps, which makes their saliency maps more clear. Nevertheless, since black-box classifiers are trained without any location information, the object locations in their high-level layers may not always correspond to the locations in raw images, leading to new issues for these explanation methods.

2.2 Perturbation-based Local Explanation

The perturbation-based methods first perturb an input according to a given mask, and then observe the new class probability of black-box classifiers. By measuring its difference to the probability of the raw input, the supporting features in the input with the class label can be located (Dabkowski & Gal, 2017). Let $\mathbf{M}$ indicate the expected mask of an image $\mathbf{I}$, where $m_{ij} \in \mathbf{M}$ indicates the relevance

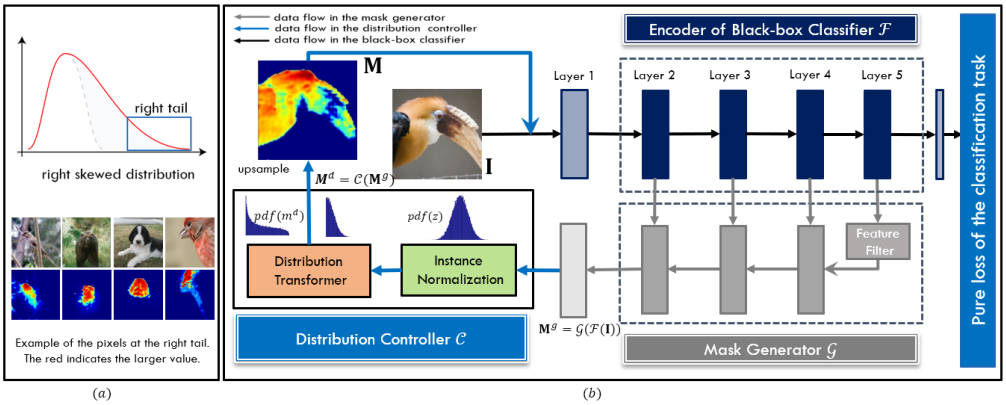

Figure 1: (a). The illustration of the benefits of using skewed distributions for explanation. (b) The framework of our mask predictor, where a distribution controller is introduced right after the mask generator. In particular, pdf stands for the probability density function.

score of the pixel within $\mathbf{I}$. Besides perturbing $\mathbf{I}$ with mask $\mathbf{M}$, these methods also introduce an alternative background image $\mathbf{A}$ to reduce the amount of unwanted evidences:

$$\phi(\mathbf{I}, \mathbf{M}) = \mathbf{I} \odot \mathbf{M} + \mathbf{A} \odot (\mathbf{1} - \mathbf{M}). \tag{1}$$

Suppose f_c indicates the probability of the predicted class c based on the black-box classifier. These method input the perturbed images to the classifier and optimize the mask with ad hoc constraints:

$$\text{argmin}_{\mathbf{M}} \ell_p(\mathbf{M}) = \text{argmin}_{\mathbf{M}} \lambda_1 \text{TV}(\mathbf{M}) + \lambda_2 \text{AV}(\mathbf{M}) - f_c(\phi(\mathbf{I}, \mathbf{M})) + \lambda_3 f_c(\phi(\mathbf{I}, \mathbf{1} - \mathbf{M})) \tag{2}$$

where TV enforces the mask to be smooth and AV aims to minimize the average of all scores. The last two terms aim to obtain discriminative scores between supporting pixels and the rest pixels.

To improve the explainability, various novel methods following the above formulation are subsequently proposed. For example, (Du et al., 2018b) regularizes the expected mask with middle-level features and optimizes the mask by reconstructing higher-level feature maps. (Fong & Vedaldi, 2017) introduces a deletion game and reformulates the problem by applying multiple masks stochastically. However, these methods need to optimize the mask for each image individually, leading to non-negligible time costs. Moreover, the large information gap between the loss of classification and these constraints increases the difficulty of tuning hyper-parameters.

3 The Proposed Method

In this section, we introduce the details of the proposed framework, as shown in Fig.1(b). It consists of an encoder inside the black-box classifier $\mathcal{F}$, a mask generator $\mathcal{G}$, and a distribution controller $\mathcal{C}$. In particular, the generator and controller together compose the mask predictor, which takes the feature maps of the black-box classifier as inputs and predicts the relevance scores for each instance. In the following part, we first introduce the mask generator. Then we present the methodology of designing our distribution controller and its implementation details. We finally introduce two metrics to evaluate the quality of explanation masks.

3.1 The Review of Mask Generator

We first describe the mask generator $\mathcal{G}$ in our framework. Specifically, it contains three bottleneck blocks and follows the U-Net architecture with the black-box classifier (Dabkowski & Gal, 2017). Each image is input into the classifier for producing feature maps at multiple layers. Then the mask generator upsamples the feature maps of lower resolution using transposed convolutions, and concatenates them with the higher-resolution feature maps. Based on multiple transposed convolution layers, the generator produces a reduced mask $\mathbf{M}^g = (\mathcal{G}(\mathcal{F}(\mathbf{I}))$ at a coarse scale and obtain the relevance score on each location. Then, upsampling based on bilinear interpolation is employed to obtain more smooth masks at the image scale. Let $\mathbf{M}$ indicate the expected mask. Denote $\mathcal{F}(\mathbf{I})$ as

the feature maps for the image $\mathbf{I}$. The image is then perturbed with the generated mask with Eq.1, and the generator can be optimized based on the constraints in Eq.2:

$$\text{argmin}_{\mathcal{G}} \ell_p(\mathbf{M}), \text{where } \mathbf{M} = \text{upsample}(\mathbf{M}^g). \tag{3}$$

In particular, a simple feature filter is pretrained to perform initial localization with respect to the predicted class. Since it is not the focus of this paper, we leave the detailed description in Appendix.

The generator module can produce explanation masks in real time. However, balancing the trade-offs between the classification loss and additional constraints (e.g., the smoothing term in Eq. 2) involves a non-trivial hyper-parameter tuning process. In addition, the framework needs to process both the perturbed $\mathbf{I} \cdot \mathbf{M}$ and the perturbed $\mathbf{I} \cdot (\mathbf{1} - \mathbf{M})$ for each image. It results in an increased GPU memory cost for training the generators with a medium batch size, e.g., 128.

3.2 The Mask Predictor

To address the above issues , we introduce a simple but effective mask predictor, which is optimized solely under the classification loss. To do this, we consider two desired properties that a good mask $\mathbf{M}$ need to satisfy with: *discrimination*, namely the high relevance scores concentrate on a small portion of supporting features while low scores are preferred on other features; *smoothness*, meaning that scores of adjacent pixels in each image are supposed to be similar.

Firstly, to produce discriminative scores in $\mathbf{M}$ without any constraint in Eq.2, we instead encode their distribution inside the mask predictor. Besides, as users require discriminative scores on each instance $\mathbf{I}$, we introduce a distribution controller $\mathcal{C}$ to guide the distribution of the relevance scores for $\mathbf{I}$ individually. Suppose $\mathbf{M}^g$ is the output of the generator on $\mathbf{I}$, and $\mathbf{M}^d$ denotes the output of $\mathcal{C}$. We obtain $\mathbf{M}^d = \mathcal{C}(\mathbf{M}^g)$. The details inside $\mathcal{C}$ will be presented in the following sections.

Secondly, to obtain a smooth mask $\mathbf{M}$, we introduce the controller right after the generator at a coarse scale (e.g., 56×56 pixels) rather than the image scale. Then we follow (Dabkowski & Gal, 2017; Du et al., 2018b) to unsample $\mathbf{M}^c$ with interpolation. Benefiting from the training with large-scale images and the unsampling operation, the output relevance scores are robust on the representative regions and smoothness can be generally guaranteed.

In short, by following the above properties to design the new strategy, we expect that the obtained predictor can facilitate the setting of hyper-parameters and remain the effectiveness of masks.

Suppose ℓ_{ce} denotes the classification loss for training the black-box classifier f, and c is the output class of the classifier. We aim to find the region that maximizes the target classification under the expected distribution. Thus, the mask predictor can be optimized only based on ℓ_{ce} as

$$\text{argmin}_{\mathcal{C},\mathcal{G}} \ell_{ce}(f(\phi(\mathbf{I}, \mathbf{M})), c), \text{ where } \mathbf{M} = \text{upsample}(\mathcal{C}(\mathbf{M}^g)). \tag{4}$$

Different from (Dabkowski & Gal, 2017), the above predictor is optimized solely under the classification loss and only involves the perturbed $\mathbf{I} \cdot \mathbf{M}$. Thus, it better mimics the black-box classifier and reduces GPU memory cost.

3.2.1 The Principles of Controller Design

Now we investigate some principles for designing the distribution controller $\mathcal{C}$.

Principle 1. We expect a right-skewed distribution of relevance scores as interpretation for each instance (see Fig.1(a)). The motivation behind is that a large portion of contributions to each decision are supposed to be held only by a small fraction of the supporting features. These supporting features, whose information should be preserved by the mask, are assigned with high scores. Under this principle, a proportion of features corresponding to the area at the right tail will be highlighted with greater scores. The distribution with a narrow peak and a longer tail is desired, which corresponds to better discrimination among the supporting pixels with different contributions.

Principle 2. We impose the monotonic mapping from the distribution controller's input $\mathbf{M}^g$ to its $\mathbf{M}^d$. The motivation is to enhance the discriminative ability of the coarse mask $\mathbf{M}^g$ but without changing its ranking of relevance scores. Of note, if the monotonicity is not guaranteed for the controller, the training produce of the mask predictor would become unstable.

3.2.2 THE DESIGNS OF CONTROLLER

The controller design includes two steps: (1) initializing a symmetric distribution for random inputs without changing their ranking; (2) transforming the distribution monotonically to a right-skewed distribution (see Fig.1(b)).

Firstly, let $z_{ij} \in \mathbf{Z}$ be the expected variable with a symmetric distribution, and $m^g_{ij} \in \mathbf{M}^g$ indicates the output at the location (i, j) of the mask generator. Instance normalization (Ulyanov et al., 2016) can be easily used to build a normal distribution. Specifically, $z_{ij} = (m^g_{ij} - \mathrm{E}[m^g_{ij}])/(\sqrt{\mathrm{Var}[m^g_{ij}]})$, where the expectation and variance are computed over the outputs of each $\mathbf{M}^g$.

Secondly, let $\mathbf{M}^d$ be the expected output of a distribution controller. We introduce two monomaniacal transformers to achieve right-skewed distributions for $m^d_{ij} \in \mathbf{M}^d$, including a basic design without any hyper-parameter, and a customized design with easy-to-set hyper-parameters.

Basic Transformer. The straightforward way of transforming a normal distribution into a right-skewed distributions is cropping variables with Rectified Linear Unit (ReLU) (Nair & Hinton, 2010). Specifically, m^d_{ij} can be obtained as

$$m^d_{ij} = \mathrm{ReLU}(z_{ij}) = \max(0, z_{ij}). \tag{5}$$

The benefit of this transformer is that, it allows a monomaniacal transformation without any hyper-parameter. Nevertheless, its half-right distribution faces a large variance and may also meet outliers, which reduces the discrimination of the sores on supporting features. Besides, this transformer is only able to estimate the relevance scores for a half of the features.

Customized Transformer. To address the issues of the basic transformer, we introduce an alternative with easy-to-set hyper-parameters. The goal is to produce scores for all features with a right tail in $(0, 1)$. Specifically, we first transform the normal distribution towards an uniform distribution based on $\mathrm{sigmoid}$ functions, and then change the skewness of the distribution based on power functions:

$$m^d_{ij} = \left(\mathrm{sigmoid}(\eta \cdot z_{ij})\right)^h, \tag{6}$$

where η is used to approach the uniform distribution, and h determines the skewness of the obtain distribution. The detailed setting of hyper-parameters will be discussed in Sec. 3.2.3.

In summary, based on two steps of the monomaniacal instance normalization and the monomaniacal transformation, we have transferred the process of tuning hyper-parameters in Eq.2 into the task of distribution design with the involved hyper-parameters in a transformer, e.g., Eq.6. The benefit is that, the effects of these involved hyper-parameters on the scores can be estimated without model optimization, and the hyper-parameters can be easily set. It is worth noting that other transformers with a monotonic mapping can also be considered, depending on users' preference. We choose the above functions for the customized transformer owing to their intuitive geometric properties.

3.2.3 THE EXAMPLE OF HYPER-PARAMETER SETTING FOR DISTRIBUTION DESIGN

We present an example of hyper-parameter setting for the customized transformer with two steps.

Firstly, we estimate the probability density function of m^d. With the probability density transformation Forbes et al. (2011), the transformed probability density function can be obtained as

$$p(m^d) = \frac{1}{\sqrt{2\pi}h\eta} \cdot \frac{1}{m(1 - m^{1/h})} \cdot \exp\left(-\frac{\left(\ln(m^{(-1/h)} - 1)\right)^2}{2\eta^2}\right), \tag{7}$$

where the superscript d is removed for clarity. The detailed proof is provided in Appendix.

Secondly, we set the hyper-parameters based on their effects on the geometry of probability density functions. We fix $h = 1$ and observe the effect of η. The corresponding probability density function is displayed in Fig.2(a). By changing η within (0.5,2.5), the density functions of $p(m^d)$ approximately change from the concave to the convex for $m^d \in (0, 1)$. Then, we set $\eta = \{0.5, 1.5, 2.5\}$ and observe the effect of h. In particular, $\eta = 1.5$ approximately leads $p(m^d)$ to an uniform distribution in Fig.2(a). Three probability density functions are displayed in Fig.2(b), Fig.2(c), Fig.2(d). Owing to the strong concavity and convexity in the first step of the transformer, $p(m^d)$ with $\eta = 0.5$ still

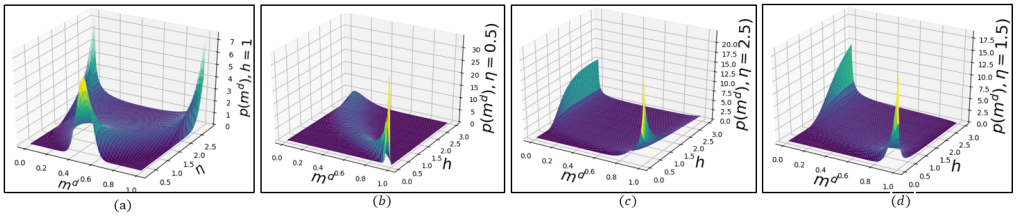

Figure 2: The probability density functions with the different settings of hyper-parameters.

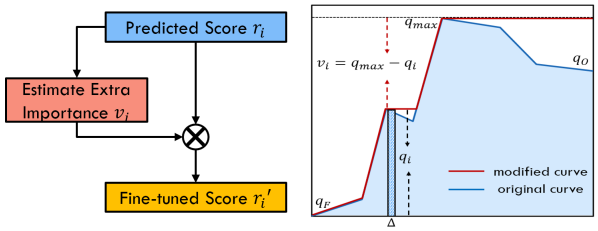

Figure 3: Left: The example of tuning scores. Right: the modification of class probabilities and the estimation of $\mathcal{M}_{\mathcal{F}}$.

faces a large variance and $p(m^d)$ with $\eta = 2.5$ continues the undesired concavity. On the contrary, benefiting from an approximated uniform distribution with $\eta = 1.5$, the distribution of $p(m^d)$ with $\eta = 1.5$ is modified from the trend of the left-skewed to the right-skewed with a small variance in Fig.2(d). Since a proportion of features corresponding to the area at the tail will be highlighted with higher scores, we fix $\eta = 1.5$ and $h = 2.5$ to obtain discriminative scores on supporting features.

3.3 Fine-tuning For Highlighting Supporting Features

So far, we have provided the strategy to facilitate the setting of hyper-parameters while remaining the desired properties of masks. The controller $\mathcal{C}$ provides the same distribution of relevance scores for different images. However, as the number of supporting pixels varies across images, these scores are supposed to be fine-tuned for each image individually.

Specifically, suppose r_i is the i-th highest score in the predicted mask. Denote q_i as the class probability of the masked image, where only the top i pixels are not perturbed, and $q_{max} = \max_i q_i$. We prefer the modified scores r_i' to follow two conditions:
(1). if $r_i > r_j$, then $r_i' > r_j'$; (2). If q_i is monotonically increasing w.r.t. i, then $r_i' \propto q_{max} - q_i$.
The former demands that the ranking of relevance scores should be remained. The later implies that the pixel is not likely to be the supporting one if its accumulated class probability already approaches the maximum.

We present a very simple technique to satisfy the conditions. Firstly, we estimate the class probability q_i with the increasing number of pixels i, and then refine the monotonicity with $q_i = \max q_{j \leq i}$. After that, we calculate the extra importance for each pixel as $v_i = (q_{max} - q_i)$. A toy example is shown in Fig.3, in which the curve of real probabilities in blue is changed to the red one for monotonicity. The final score can be calculated as $r_i' = r_i \times v_i$. For efficiency, we can uniformly sample a limited number of i for estimating the above probabilities and infer the remaining with linear interpolation.

3.4 Generalized Evaluation Metrics

Evaluating the quality of saliency maps based on a heuristical segmentation of images will reduce the fairness of comparisons. It is understandable that the segmentation will be significantly affected by thresholds (Fong & Vedaldi, 2017). To address this issue, we introduce two generalized metrics to evaluate the comprehensive quality in terms of both the faithfulness and the explainability.

Faithfulness. The explanation is expected to accurately replicate the models behaviour. Here we rely on the smallest sufficient region of the image that alone allows a confident and consistent classification. Compared with smallest destroying regions, it encourages to find explanations more consistent with the classifiers training distribution (Chang et al., 2019). Since it is subjective to decide how much confidence is preferred for a specific decision, we extend the evaluation into a more general case by taking advantage of the ranking of relevance scores.

Specifically, we first estimate the class probability of fully-perturbed images q_F. We reduce the percent of perturbed pixels at intervals of Δ with the decremental ranking of scores and calculate their probabilities q_is, until q_O is free of any perturbation. The area under the probability vs. the percent of pixels curve is used to evaluate the faithfulness, which is integrated with a finite sum:

$$\mathcal{M}_{\mathcal{F}} = \sum_i (\frac{q_i}{q_O - q_F} \cdot \Delta) \times 100\%. \tag{8}$$

For illustration, an example is displayed in Fig.3, where $\mathcal{M}_{\mathcal{F}}$ indicates the area in blue. Besides, an extra discussion between this metric and perturbation-based methods are provided in Appendix.

Explainability. For quantitative evaluation of explainability, we can use the extra information such as bounding boxes for weakly-supervised object localization or the pointing game (Du et al., 2018b). However, the former still faces the issue of the choice of thresholds; the later meets a large bias, since it represents the quality of the scores of all pixels only based on one pixel.

Thus, we regard relevance scores as the results of retrieval tasks. Specifically, we denote the precision as the fraction of the pixels retrieved within bounding boxes. Recall denotes the fraction of the within-bounding-box pixels that are successfully retrieved. By computing a precision P_i and recall R_i at each position in the ranked scores of pixels, we can evaluate the explainability of scores by the area under the precision-recall curve:

$$\mathcal{M}_{\mathcal{I}} = \sum_i P_i \cdot (R_i - R_{i-1}) \times 100\%. \tag{9}$$

4 EXPERIMENTS

This section evaluates the effectiveness of the proposed method. We firstly estimate the ability of mimicking black boxes with the first metric. Then we evaluate the explainability through both the second metric and pointing game. Finally, we display masks for visual comparison.

We perform experiments on ImageNet, and ResNet50 (He et al., 2016) is used as the black-box classifier. We build the mask generator with three bottleneck blocks, which takes 7×7 feature maps as the low-level input, and predicts the mask at the 56×56 in size. We use a two-stage scheme to train the mask predictor. We first train the feature filter based on 250,000 images sampled from the training set, and then optimize other parts of the mask predictor with the batch size of 64. Of note, no ground truth is introduced and only the outputs of the classifier are utilized. We use Adam for 10 epochs with the initialized learning rate of 10^{-2}. We apply step decay, and reduce the learning rate by half every three epochs. In addition, during training stage, 50% of cases the image $\mathbf{A}$ is the Gaussian blurred version of $\mathbf{I}$ with a variance of 10. The remainder of cases, $\mathbf{A}$ is set to a random colour image with the addition of a Gaussian noise. Besides the proposed *Basic Transformer* (BF) and *Customized Transformer* (CF), the following methods are used for comparison: (1) Mask Generator (MGnet) (Dabkowski & Gal, 2017), (2) Meaningful Perturbation (MPert) (Fong & Vedaldi, 2017), (3) Grad CAM (Selvaraju et al., 2017), (4) Grad CAM++ (Chattopadhay et al., 2018), (5) Vanilla Gradient (V-Grad) (Simonyan et al., 2014), (6) Smoothness-Gradient (SM-Grad) (Smilkov et al., 2017), (7) Integrated Gradient (IT-Grad) (Sundararajan et al., 2017). In particular, all perturbation-based methods apply the same strategy for adding noise. For the hyper-parameters in the compared methods, we follow the setting in their papers for a fair comparison.

4.1 FAITHFULNESS WITH QUANTITATIVE EVALUATION

We sample 10,000 images from the validation set to compose the testing set and calculate the faithfulness score $\mathcal{M}_{\mathcal{F}}$. According to the definition of $\mathcal{M}_{\mathcal{F}}$, a larger value means the method can better estimate the contributions of pixels on specific decisions. We set $\Delta = \frac{1}{32}$ to sum up their probabilities. To obtain the ranking of pixels with zero scores, we simply add a smoothed mask over the

Table 1: Metric of $\mathcal{M}_{\mathcal{F}}$.

Method	Average
Ours(BF)	69.35
Ours(CF)	**71.35**
MGnet	64.93
MPert	68.93
Grad CAM	69.71
Grad CAM++	67.92
V-Grad	30.40
SM-Grad	43.25
IT-Grad	39.43

Table 2: Metric of $\mathcal{M}_{\mathcal{I}}$.

Method	Average
Ours(BF)	**84.31**
Ours(CF)	84.02
MGnet	83.16
MPert	78.21
Grad CAM	77.56
Grad CAM++	83.33
V-Grad	65.30
SM-Grad	71.91
IT-Grad	66.46

Table 3: Metric of $\mathcal{M}_{\mathcal{PG}}$.

Method	Average
Ours(BF)	87.41
Ours(CF)	85.90
MGnet	**88.63**
MPert	84.25
Grad CAM	76.72
Grad CAM++	83.42
V-Grad	76.78
SM-Grad	87.63
IT-Grad	81.94

original one with a tiny weight. We perturb images with their Gaussian blurred version based on their masks and estimate the average faithfulness.

The results of average $\mathcal{M}_{\mathcal{F}}$ are listed in Tab.1. Based on the results, the following observations can be obtained. Firstly, the former six methods outperform V-Grad, SM-Grad, and IT-Grad with a large gap. It is understandable that, these three methods only search sensitive pixels with gradients, the pixels with high scores will discretely appear in each image. As a result, these methods become harder to gather the information in a local region and reach a high class probability. Secondly, Grad CAM is slightly better than Grad CAM++. The possible reason is that, although Grad CAM ++ can localize multiple objects in an image if the image contains multiple occurrences of the same class, it also increases the possibility to generate the high scores at background and makes the pixels with high scores separated. Thirdly, our methods outperform most compared methods, and obtain much higher performance than MGnet. Since all three methods apply the same architecture of mask generators, it demonstrates the effectiveness of fitting the relevance scores towards right-skewed distributions. Finally, by constraining the tail with a small variance in (0,1), Ours(CF) improves the ranking of supporting pixels and outperforms Ours(BF).

4.2 Explainability with Quantitative Evaluation

4.2.1 Weakly Supervised Object Localization

Below we evaluate the explainability by applying the generated masks to weakly supervised object localization tasks. The second metric $\mathcal{M}_{\mathcal{I}}$ in Eq.9 is used for this task. We resize and crop bounding boxes to the size of 224×224, leading to the same size of test images. The experiments are performed on a subset of validation set, which contains 10,000 images with bounding box annotations.

The results of average $\mathcal{M}_{\mathcal{I}}$ are listed in Tab.2. From this table, we have the following observations. Firstly, Grad CAM ++ obtained much better performance than Grad CAM. It is understandable that the former is able to detect multiple objects in the image and assign them the high relevance scores. Secondly, by replacing all constraints with a simple distribution controller, our methods outperform MGnet with a small gap. Besides, benefiting from the training with large-scale images, their predicted high relevance scores focus on objects more robustly, and leads to better performance than MPert. Thirdly, the last three gradient-based methods perform worse than others. The possible reason is that, gradients are insensitive to the smooth supporting regions, which makes these regions ignored and reduces the performances.

4.2.2 Pointing Game

Now we evaluate the explainability with pointing games. Specifically, the maximum point is first extracted from each generated mask. Then according to whether the maximum point falls in one of the ground truth bounding boxes or not, a hit or a miss is counted. The localization accuracy of the pointing game for each object category is defined as: $\mathcal{M}_{\mathcal{PG}} = \frac{\#Hits}{\#Hits+\#Misses} \times 100\%$. This process is repeated for all categories and the results are averaged as the final accuracy.

The average results of all compared methods are listed in Tab. 3. From the results of this table, the following observations can be obtained. Firstly, SM-Grad and IT-Grad obtain comparable performance to the former six methods and outperform V-Grad. The main reason is that, this metric

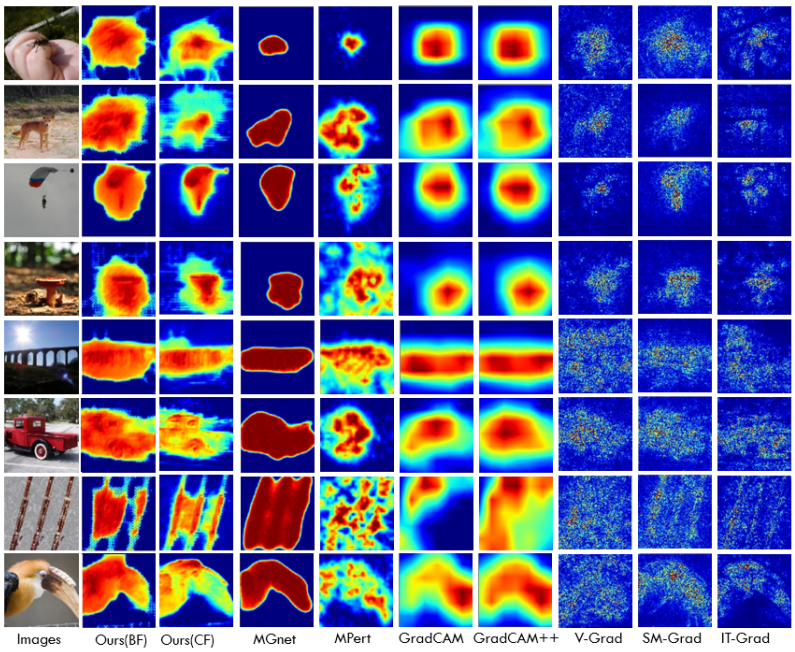

Figure 4: Saliency maps of different methods. More examples are displayed in Appendix.

only considers the localization precision of the top pixel rather than a group of pixels. By removing noise in gradients to some degree, these two methods are able to locate the most important pixel. Secondly, although our methods remove the smoothness constraint, they still obtain high accuracy of locating the most important pixel. One possible reason is that, by optimizing the predictor with training data, a large number of noisy features can be removed, leading to the robust estimation of scores. Thirdly, MGnet obtains the best performance. It means that, by enforcing the background towards a low class probability, the high scores on the background pixels are generally prevented.

4.3 Explainability with Visualization

4.3.1 Comparison to baseline methods

To visually demonstrate the explainability, we show the generated saliency maps of different methods, where the red denotes the high score and the blue indicates the low score. For a fair comparison, no fine-tuning is used for the proposed methods. We randomly sample images from ImageNet and show their masks in Fig.4. More examples are displayed in Fig.6 in Appendix.

From this figure, we have the following observations. Firstly, the gradient-based methods generally results in more noise outside the objects. Secondly, although the high-level feature maps are used to build saliency maps, Grad CAM and Grad CAM++ may still miss the supporting regions of objects, such as the last two examples in Fig.4. Thirdly, although MGnet generally locates the positions of objects, it may lead to redundancy high scores when multiple objects exist. Finally, with an easy setting of hyper-parameters, Ours(CF) can obtain more discriminative saliency masks than Ours(BF). It demonstrates the effectiveness of introducing a small-variance tail within $(0, 1)$.

4.3.2 The effect of fine-tuning

In this section, we aim to demonstrate the effectiveness of fine-tuning. As we mentioned, constraining different images with the same distribution may lead to redundant scores on unimportant pixels, especially when the supporting pixels only take a small part of the image. Thus, we perform the fine-tuning on the obtained masks. Fig.5 shows same examples of our masks before and after the fine-tuning operation. The masks of MGnet are also displayed for comparison. More examples of this task can be found in Fig.7 in Appendix.

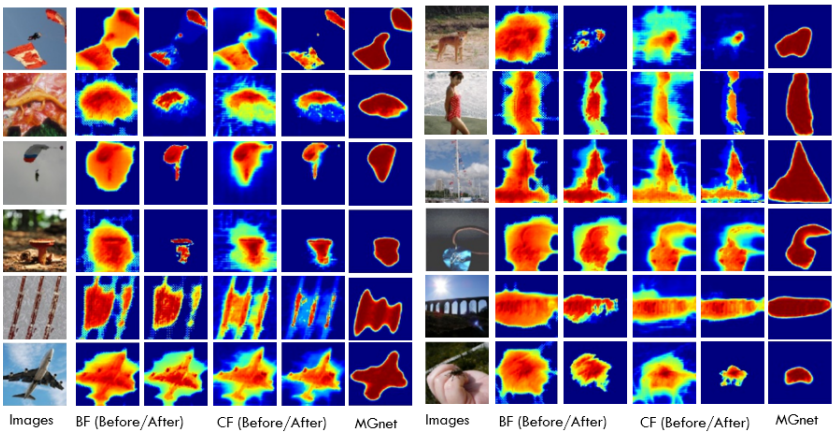

Figure 5: Some saliency maps before and after fine-tuning. More examples are shown in Appendix.

From the figure, we observe that although the obtained raw masks are generally targeted at objects with discriminative scores, there are still many redundant relevance scores on unimportant pixels. By weighting them with the extra importance in Sec. 3.3, we can remove the noise and highlight the supporting region which has high relevance to the predicted class.

5 Conclusion and Further Work

This paper presents a simple but effective mask predictor to provide local explanations for DNNs. Specifically, we replace the ad hoc constraints with a distribution controller, and integrate it with a mask generator to directly guide the distribution of relevance scores. The benefit is that, it facilitates the setting of involved hyper-parameters, and enables discriminative scores over supporting features. The experimental results demonstrate that our method outperforms others in terms of faithfulness and explainability. Meanwhile, it also provides effective saliency maps for explaining each decision.

There are some aspects needing further investigations. Firstly, other different distributions may be explored for guiding the distribution of relevance scores. Secondly, although this paper provides an intuitive comparison of the transformed distributions for setting hyper-parameters, a more quantitative analysis on the proportion of features at the tail could be studied. Thirdly, various advanced techniques, such as Dropout, can be used for further improving the smoothness of masks.

References

Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. *Seventh International Conference on Learning Representations (ICLR)*, 2019.

Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847. IEEE, 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 *Advances in neural information processing systems (NIPS)*, pp. 2172–2180, 2016.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 6967–6976, 2017.

Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *arXiv preprint arXiv:1808.00033*, 2018a.

Mengnan Du, Ninghao Liu, Qingquan Song, and Xia Hu. Towards explanation of dnn-based prediction with guided feature inversion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2018b.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3429–3437, 2017.

Catherine Forbes, Merran Evans, Nicholas Hastings, and Brian Peacock. Statistical distributions. 2011.

Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems (TNNLS)*, 28(10):2222–2232, 2016.

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 (CVPR)*, pp. 770–778, 2016.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

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

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML)*, pp. 807–814, 2010.

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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *ICLR Workshop*, 2014.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *ICML workshop*, 2017.

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

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3319–3328. JMLR. org, 2017.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

A APPENDIX

A.1 THE FEATURE FILTER IN THE MASK GENERATOR

The purpose of the feature filter is to attenuate spatial locations which contents do not correspond to the selected class. Denote $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ as the output of the last convolution layer of the classifier, where H,W C indicate the height, the width, and the number of channels of its feature maps, respectively. The output of filter $\mathbf{Y} \in \mathbb{R}^{H \times W \times C}$ at the spatial location i, j is formulated as

$$\mathbf{Y}_{ij} = \mathbf{X}_{ij} \text{sigmoid}(\mathbf{X}_{ij}^{\mathrm{T}} \mathbf{C}_s), \tag{10}$$

where $\mathbf{C}_s \in \mathbb{R}^{1 \times C}$ is the embedding of the class c. For efficient optimization, (Dabkowski & Gal, 2017) introduces noise on real labels. Specifically, gradient-based optimizers are employed to maximize $\text{mean}(\text{sigmoid}(X_{ij}^T C_k))$, $k = s$ and minimize $\text{mean}(\text{sigmoid}(\mathbf{X}_{ij}^{\mathrm{T}} \mathbf{C}_s))$, $k \neq s$, iteratively.

A.2 A BRIEF INTRODUCTION OF THE EXPERIMENTAL SETTING

Below we present a brief description on the experimental settings of compared methods.

- Mask Generator (Dabkowski & Gal, 2017), which introduces the multiple constraints into the objective functions for training the predictor. For this method, we use the same setting in our method and using its default hyper-parameters in its publicly available codes.
- Meaningful Perturbation (Fong & Vedaldi, 2017), which performs meaningful image perturbations and directly optimizes masks with designed constraints. For comparison, we use the same kind of perturbations in our method, and apply an Adam optimizer with the learning rate of 0.1 for optimization. The iterations is set to 300.
- Grad CAM (Selvaraju et al., 2017), which unsamples the saliency maps based on the gradient-weighted high-level feature maps. We use the last convolutional layers to build its coarse saliency map and upsample it to the image scale as the final saliency map.
- Grad CAM++ (Chattopadhay et al., 2018), which conducts a weighted combination of the positive partial derivatives to generate a visual explanation. We apply Grad CAM ++ with exponential functions for efficiency.
- Vanilla Gradient (Simonyan et al., 2014), the basic method that uses the gradients of the raw images as saliency maps. To improve the visualization, we crop outliers and normalize the scores to $[0, 1]$.
- Smoothness-Gradient (Smilkov et al., 2017), which removes noise by adding noise to images. We apply 20% noise as suggested and set the sample size to 50. Similarly, we crop outliers and normalize the scores.
- Integrated Gradient (Sundararajan et al., 2017), which combines the implementation invariance of gradients along with the sensitivity. We introduce black images as the baseline and set 200 as the number of steps in its Riemman approximation of the integral. We use the same way of improving visualization for saliency maps.

A.3 THE PROOF OF EQ.7.

According to the probability density transformation Forbes et al. (2011), the transformed probability density function can be obtained based on that of the original variable:

$$p(m) = p_z(g_z^{-1}(m)) \cdot |\frac{\partial g_z^{-1}(m)}{\partial m}|, \tag{11}$$

where $p_z(\cdot)$ is the probability density function of the original variable z, and $g_z^{-1}(m)$ is the inverse function of m on z. Since $m = g_z(z) = (\frac{1}{1+\exp(-\eta z)})^h \in (0, 1)$ in Eq.6, we obtain

$$z = g_z^{-1}(m) = -\frac{1}{\eta} \ln(m^{-1/h} - 1), \tag{12}$$

where $(m^{-1/h}-1)>0$. Besides, we obtain:

$$\begin{aligned}|\frac{\partial g_z^{-1}(m)}{\partial m}| &= \frac{1}{\eta}\cdot\frac{1}{m^{(-1/h)}-1}\cdot(\frac{1}{h}m^{(-1/h)-1}) \\ &= \frac{1}{\eta h}\cdot\frac{1}{m}\cdot\frac{m^{(-1/h)}}{m^{(-1/h)}-1} \\ &= \frac{1}{\eta h}\cdot\frac{1}{m(1-m^{(1/h)})}\end{aligned} \tag{13}$$

Recall the probability density function of a normal distribution is

$$p_z(z) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}). \tag{14}$$

where $\sigma=1$ in standard normal distributions. By substituting Eqs.12-14 into Eq.11, we obtain

$$p(m) = \frac{1}{\sqrt{2\pi}h\eta}\cdot\frac{1}{m(1-m^{(1/h)})}\exp(-\frac{\left(\ln(m^{(-1/h)}-1)\right)^2}{2\eta^2}), \tag{15}$$

which completes the proof.

A.4 The analysis on the relationship between the proposed metric and the perturbation-based methods.

We provide a new view to discuss the relationship between the proposed metric and wildly used perturbation-based methods with the mask involved. Recall that in most perturbation-based methods, we aim to maximize the probability of the target class of the perturbed image:

$$\mathrm{argmin} f_c(\phi(\mathbf{I},\mathbf{M})), \tag{16}$$

where the pixels with higher scores in masks are supposed to be more important. By decomposing the above mask into multiple ones with one-hot coded mask M_i, Eq.16 is equal to

$$f_c(\phi(\mathbf{I},\sum_{i=0}^{N}\alpha_i\mathbf{M}_i)) \tag{17}$$

where α_i denotes the relevance score with $\alpha_i>\alpha_{i+1}$. Suppose β_i indicate some positive weights and denote $\alpha_i=\sum_{j=i}^{N}\beta_j$, the above equation can be further transformed into

$$f_c(\phi(\mathbf{I},\sum_{i=0}^{N}(\sum_{j=i}^{N}\beta_j)\mathbf{M}_i)) = f_c(\phi(\mathbf{I},\sum_{i=0}^{N}(\beta_i+...+\beta_N)\mathbf{M}_i)). \tag{18}$$

Recall the area under the probability vs. the number of pixels, which can be formulated as

$$\sum_{j=0}^{N}(\beta_j f_c(\phi(\mathbf{I},\sum_{i=0}^{j}\mathbf{M}_j))) = \beta_0 f_c(\phi(\mathbf{I},\mathbf{M}_0))) + \beta_1 f_c(\phi(\mathbf{I},\mathbf{M}_0+\mathbf{M}_1)) + ... \tag{19}$$

We observe that Eq.18 tends to be a linear approximation of Eq.19. Specifically, two of them become equal when $f_c(\cdot)$ and $\phi(\cdot)$ are linear. According to Eq.19, the larger values on the former β_i means the users pay more attention to the top features. According to Eq.18, it also implies the high scores should be much larger than others and highlight important features. Thus, to evaluate the effectiveness of perturbation-based methods, the proposed metric tends to be more convincing.

A.5 Saliency Maps of different methods.

More saliency maps generated by different methods are displayed in Fig.6.

A.6 Saliency Maps of Fine-tuning.

Saliency maps of the proposed methods before and after fine-tuning are displayed in Fig.7. In addition, the results of MGnet are displayed for comparison.

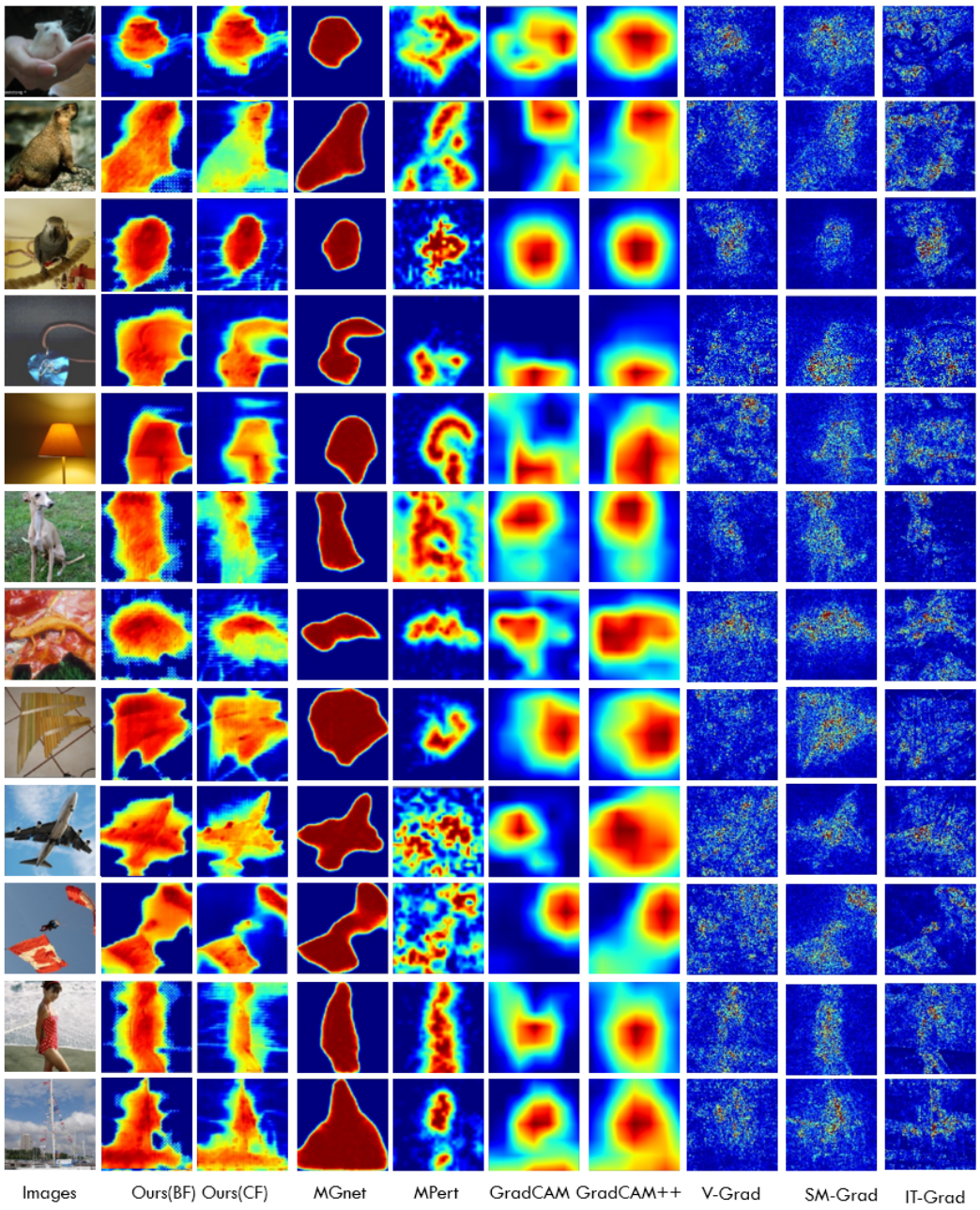

Figure 6: Example of the saliency maps of different methods.

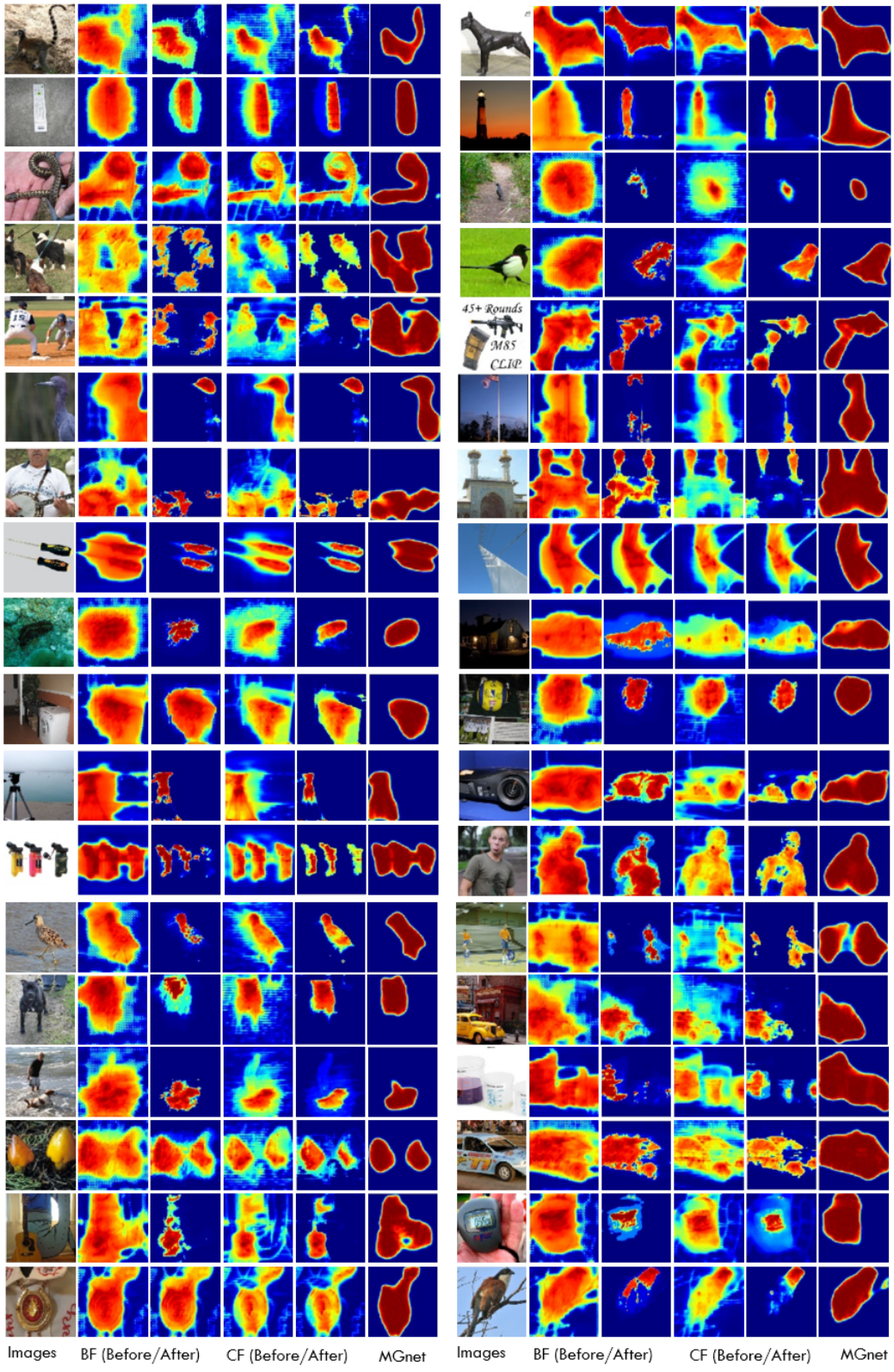

Figure 7: Saliency maps of the proposed methods before and after fine-tuning.