Differentiated Explanation of Deep Neural Networks with Skewed Distributions

Weijie Fu, Meng Wang, *Fellow, IEEE,*
Mengnan Du, Ninghao Liu, Shijie Hao, and Xia Hu, *Member, IEEE*

Abstract—Over the last decade, deep neural networks (DNNs) are regarded as black-box methods, and their decisions are criticized for the lack of explainability. Existing attempts based on local explanations offer each input a visual saliency map, where the supporting features that contribute to the decision are emphasized with high relevance scores. In this paper, we improve the saliency map based on differentiated explanations, of which the saliency map not only distinguishes the supporting features from backgrounds but also shows the different degrees of importance of the various parts within the supporting features. To do this, we propose to learn a differentiated relevance estimator called DRE, where a carefully-designed distribution controller is introduced to guide the relevance scores towards right-skewed distributions. DRE can be directly optimized under pure classification losses, enabling higher faithfulness of explanations and avoiding non-trivial hyper-parameter tuning. The experimental results on three real-world datasets demonstrate that our differentiated explanations significantly improve the faithfulness with high explainability. Our code and trained models are available at https://github.com/fuweijie/DRE.

Index Terms—deep neural networks, local explanation, relevance scores, differentiated saliency maps

1 Introduction

Deep neural networks (DNNs) have achieved high accuracies in a wide range of fields, such as image recognition [11], and natural language processing [10]. However, they often lack meaningful explanations about how specific decisions are made and are regarded as black-box methods. In particular, the explanation should take both faithfulness and explainability into account. The faithfulness estimates the fidelity between the explanation and the decision behavior of original DNNs, and the explainability quantifies how easy it is to understand the explanation for humans.

Local explanation methods are proposed to address this issue. They provide users an understandable rationale for each specific decision with a visual saliency map, where the relevance score of each feature indicates its contribution to the decision. For high faithfulness, the supporting features contributing to increase the probability of the target class are supposed to obtain high scores, and the remaining features regarded as backgrounds are expected to get almost zero scores. In particular, gradient-based explanations compute the partial derivative of the class probability with respect to input features via back-propagation [2], [20]. Besides, perturbation-based explanations aim to find the smallest region, which allows a confident decision directly or prevents a confident decision once being removed [8]. By applying various ad hoc constraints on the region and lowering the contributions of intricate supporting features, they maintain faithfulness and improve explainability.

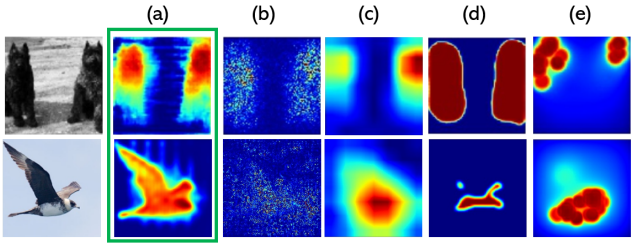


Fig. 1. Comparison of different saliency maps: (a) The proposed DRE, (b) Vanilla Gradient [20], (c) Grad-CAM++ [2], (d) Mask Generator [4], and (e) Extreme Perturbation [7].

To provide better explanations of the decisions, differentiated saliency maps are preferred. That is, *the strong supporting features that significantly contribute to the probability of the target class are highlighted with very high scores, the weak supporting features that slowly increase the probability obtain lower scores, and the other features regarded as the background have almost zero scores*. Based on the differentiated explanations, users not only can locate the whole set of supporting features but also figure out which parts of them are more important than the others. For illustration, two examples are shown in Fig.1(a), which not only capture the shapes of the whole animals but also provide detailed insights that their heads contribute more than the remaining parts.

However, the existing local methods fail to produce the differentiated explanations. For example, instead of directly addressing the basic question "what makes this image belong to the target class", the gradient-based methods answer the question "what makes this instance more or

- *W. Fu, M. Wang are with the School of Computer and Information, Hefei University of Technology, Hefei, 230009, China, and Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology) (e-mail: {fwj.edu, eric.mengwang}@gmail.com).*
- *S. Hao is with the School of Computer and Information, Hefei University of Technology, 230009, China (e-mail: hfut.hsj@gmail.com).*
- *M. Du, N. Liu, X. Hu are with the School of Computer Science and Engineering, Texas A&M University, TX 77840, U.S.A. (e-mail: {dumengnan, nhliu43}@tamu.com, and hu@cse.tamu.edu).*
- *M. Wang is the corresponding author.*

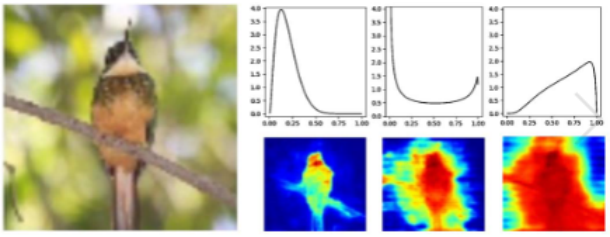


Fig. 2. The saliency maps obtained from the right-skewed distribution controller to the left-skewed distribution controller, illustrating the benefit of the right-skewed distribution for human-friendly explanations.

less similar to the target class" [16], leading to noisy results within the same region (Fig.1(b)). [2], [19] propose to create saliency maps by combining the gradients with the corresponding feature maps at high-level layers. However, they ignore the fine-grained information within low-level layers and bring coarse saliency maps (Fig.1(c)). In addition, the perturbation-based methods are formulated to highlight the supporting features directly, which ignore the different degrees of importance of these features [4] (Fig.1(d)). Recently, some explanation methods introduce soft ad hoc constraints and data augmentation techniques to improve their saliency maps [6], [24]. Nevertheless, they either significantly increase the number of iterations for optimizing each saliency map, or require users to carefully tune hyper-parameters to trade-off the constraints and the classification loss, leading to non-negligible costs. Although [7] introduces extreme perturbations with hard constraints to ease the setting of hyper-parameters, its saliency maps ignore some parts of the supporting features and still lose the differentiation on the detected supporting features (Fig.1(e)).

In this paper, we propose to learn a Differentiated Relevance Estimator (DRE) to construct differentiated explanations. Leveraging the quantitative observations found in [2], [7] that the occlusion of 5% (25%) pixels in natural images can bring nearly 50% (90%) drop in classification confidence, we present distribution controllers to guide the relevance scores towards right-skew distributions [3], so as to improve the consistency between the scores of input features and the actual contributions. Our qualitative experimental analysis on the skewness of distributions additionally shows the effectiveness of the right-skewed distribution for building human-friendly explanations, as displayed in Fig.2. We introduce the detailed setting of the controller by establishing the connections between its input and output distributions and then integrate it with a trainable mask generator to build the final estimator. Benefiting from the controller, we directly optimize DRE under classification losses, which avoids all ad hoc constraints and non-trivial hyper-parameter tuning. We further discuss a simple trick to improve saliency maps based on the ranking of relevance scores itself, which offers DRE more flexibility to address the various proportions of supporting features across instances.

The main contributions of our work are as follows.

- We introduce differentiated explanations and propose a novel relevance estimator DRE by integrating a distribution controller with a trainable mask generator. We develop a practical controller to guide relevance scores towards the desired right-skewed distributions, where the involved hyper-parameters can be easily set.
- We introduce classification losses to train DRE directly. It avoids the non-trivial hyper-parameter tuning on ad hoc constraints and also significantly improves the faithfulness of explanations.
- We empirically demonstrate the effectiveness of the above innovations with targeted ablation studies. Besides, the experimental comparison to other methods shows that DRE not only obtains better quantitative performance but also provides differentiated saliency maps for human-friendly explanations.
- We extend DRE with simple tricks with post hoc tuning. The results show that DRE can easily benefit from itself and be adaptive to different images.

2 Related Work

Gradient-based methods. Gradient-based methods leverage back-propagation to track information from the DNN's output back to its input [20]. 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, the saliency maps based on the naive gradients are visually noisy and hard to understand. To address this issue, Smooth Grad [22] reduces the visual noise by introducing noise to inputs repeatedly, and Integrated Grad [24] estimates the global contribution of each feature rather than the local sensitivity. Guided back-prorogation [23] modifies the gradients of ReLU functions by discarding negative values at the back-propagation process. Besides, recent methods propose to create saliency maps by combining the gradients with the corresponding feature maps. For example, Grad CAM [19] and Grad CAM ++ [2] take advantage of high-level feature maps to make saliency maps cleaner. Nevertheless, they inevitably sacrifice the detailed estimation of the contributions of input features and lead to coarse saliency maps.

Perturbation-based methods. Perturbation-based methods optimize the saliency map of each decision by perturbing its input features and observing the change in the output of DNNs. For example, [8] designs a preservation game to find the smallest region that significantly increases the probability of the target class. The authors also design a deletion game by preventing DNNs from recognizing objects. To improve the explainability, [6] regularizes saliency maps with middle-level feature maps and optimizes them by reconstructing higher-level feature maps. [7] further introduces extreme perturbations with a hard constraint on saliency maps, aiming to avoid the hyper-parameter tuning on soft ad hoc constraints. Nevertheless, to obtain a high-quality explanation, the above methods demand hundreds of iterations for optimizing the saliency map for each image, leading to non-negligible time costs. Recently, [4] proposes an efficient method for real-time saliency detection, which utilizes a trainable network to generate saliency maps.

Model-agnostic methods. To make the explanations compatible with more types of data and black-box classifiers, model-agnostic methods are proposed, such as LIME

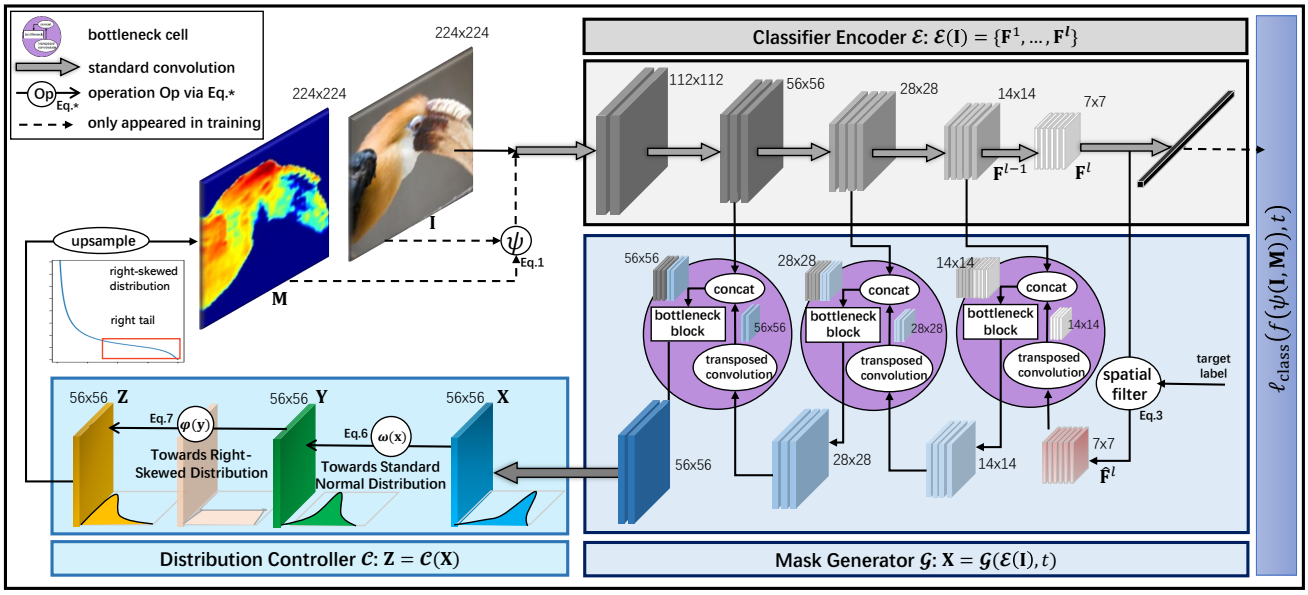


Fig. 3. The framework of our differentiated relevance estimator, where a distribution controller $\mathcal{C}$ is introduced right after the mask generator $\mathcal{G}$. For each instance $\mathbf{I}$, $\mathcal{G}$ takes the feature maps of the neural network $\mathcal{E}(\mathbf{I})$ as inputs and feeds the obtained mask $\mathbf{X}$ into the controller $\mathcal{C}$. The detailed process flow inside the mask generator $\mathcal{G}$ can be found in Sec.3.2.2. Then $\mathcal{C}$ guides the relevance scores towards the right-skewed distribution for a differentiated mask through $\mathbf{X}\rightarrow\mathbf{Y}\rightarrow\mathbf{Z}$. The final mask $\mathbf{M}$ with the original size is obtained via upsampling. In addition, we annotate the spatial sizes of feature maps and display the expected distributions of the scores within the controller.

[17], SHAP [15], and Anchor [18]. In particular, LIME employs more interpretable linear models to approximate the decisions of black-box classifiers. It assumes that each explanation can be derived from the points randomly generated around the neighborhood of the instance and their proximity measures. SHAP further introduces a united framework for interpreting decisions based on Shapley values and builds its connection to LIME. Nevertheless, these methods generally take much larger time costs to converge. For example, LIME takes around 10 minutes to explain each decision of Inception networks, and Shapley values take a more considerable time cost to compute [17].

3 Differentiated Explanation

3.1 Problem Statement

In a multi-class classification task, suppose a DNN classifier f is already trained over a training set. For each instance $\mathbf{I}$, local explanation aims to find out the contributions of its input features to the probability of the target class that we want to interpret. Take image classification as an example, where $I_{i,j}$ denotes to the pixel of $\mathbf{I}$ at the location of i, j. The corresponding local explanation is represented by a same-size mask[1] $\mathbf{M}$, in which each relevance score $M_{i,j}\in[0,1]$ represents the contribution of $I_{i,j}$ for the target probability. To improve explanations, differentiated masks are preferred.

We first analyze the perturbation-based methods [4], [8] in Sec. 3.2 and then introduce our method for differentiated explanations with skewed distributions in Sec. 3.3 - Sec. 3.5. Some important notations used in Sec. 3 are listed in Tab.1.

1. In this paper, we do not distinguish saliency maps and masks, as both indicate the permutation of relevance scores of an instance.

TABLE 1
Notations and definitions.

Notation	Definition
$\mathcal{E}$	The encoder of the classifier.
$\mathcal{G}$	The mask generator after the encoder.
$\mathcal{C}$	The distribution controller after the generator.
t	The label of the target (predicted) class.
l	The number of convolutional layers.
$\mathbf{I}$	The notation for images.
$\mathbf{B}$	The notation for background images.
$\mathbf{M}$	The notation for saliency maps or masks.
i, j, k	The symbols used for indexes.
$\mathbf{F}^k$	The feature maps at the k-th layer ($1 \leq k \leq l$).
$\mathbf{V}_k$	The embedding vector of the k-th class.
$\hat{\mathbf{F}}^l$	The last feature maps after the spatial attenuation.
$\mathbf{X}$	The output of the mask generator $\mathcal{G}$.
$\mathbf{Y}$	The intermediate variable inside the controller $\mathcal{C}$.
$\mathbf{Z}$	The output of the distribution controller $\mathcal{C}$.
$\ell.$	The symbol for different losses.
$\psi(\cdot,\cdot)$	The image perturbation with Eq.1.
$\omega(x)$	The instance normalization on x with Eq.6.
$\varphi(y)$	The transformation function on y with Eq.7.
$p(z)$	The probability density function of the variable z.
$\mathbb{E}[x]$	The expectation of the variable x.
$\mathbb{V}[x]$	The variance of the variable x.
η, h	The parameters inside $\mathcal{C}$, set based on $p(z)$.

3.2 Perturbation Analysis

3.2.1 *Formulation of Perturbation-based Methods*

To find supporting pixels, these methods perturb $\mathbf{I}$ according to an initialized mask $\mathbf{M}$ and introduce an alternative background image $\mathbf{B}$ to reduce the amount of unwanted evidence. Specifically, the perturbation is defined as

$$\psi(\mathbf{I}, \mathbf{M}) = \mathbf{I} \odot \mathbf{M} + \mathbf{B} \odot (\mathbf{1} - \mathbf{M}), \quad (1)$$

where $\odot$ denotes the Hadamard product. Then these methods feed the perturbed image into the classifier and optimize the mask to locate the supporting pixels that increase the

probability of the target class [4]. Specifically, let t denote the label of the target class, and $f_t(\psi(\mathbf{I},\mathbf{M}))$ is the corresponding class probability of the above perturbed image. The objective ℓ_{pert} of these methods can be formulated as

$$\begin{aligned}\text{argmin}_{\mathbf{M}} -f_t(\psi(\mathbf{I},\mathbf{M})) + \lambda_{\text{bg}} f_t(\psi(\mathbf{I},\mathbf{1}-\mathbf{M})) \\ + \lambda_{\text{av}}\Theta_{\text{av}}(\mathbf{M}) + \lambda_{\text{tv}}\Theta_{\text{tv}}(\mathbf{M}),\end{aligned} \tag{2}$$

where $-f_t(\psi(\mathbf{I},\mathbf{M}))$ encourages the supporting pixels to obtain high relevance scores, and $f_t(\psi(\mathbf{I},\mathbf{1}\text{-}\mathbf{M}))$ aims to avoid the supporting pixels being regarded as the background. Besides, the constraint $\Theta_{\text{av}}(\cdot)$ is used to minimize the area of the mask, and $\Theta_{\text{tv}}(\cdot)$ enforces it to be smooth.

3.2.2 *Real-time Mask Generator*

The iterative optimization for the above problem results in a considerable time cost for each test image. Thus, a mask generator $\mathcal{G}$ that produces real-time saliency maps is proposed in [4]. The simplified architecture is displayed at the bottom-right of Fig.3, which consists of a class-related spatial filter, three bottleneck cells, and a standard convolutional layer.

During the mask generation, the classifier first produces raw feature maps $\{\mathbf{F}^k\}_{k=1}^l$ at multiple layers for each image based on the encoder $\mathcal{E}$ and obtains its label t=$\text{argmax}_k f_k$. Then for the last feature maps $\mathbf{F}^l$, the spatial filter uses its class-related embedding vectors to attenuate the spatial locations whose feature vectors are dissimilar to the embedding of the target class. Let $\mathbf{V}_t$ be the above embedding vector. The output of the spatial filter at the location i,j denoted as $\hat{\mathbf{F}}^l_{ij}$ is calculated as

$$\hat{\mathbf{F}}^l_{ij} = \mathbf{F}^l_{ij}\text{sigmoid}({\mathbf{F}^l_{ij}}^{\text{T}}\mathbf{V}_t). \tag{3}$$

where $\mathbf{F}^l_{ij}$ denotes the feature vector of $\mathbf{F}^l$ at the location i,j. The first bottleneck cell then upsamples the filtered maps $\hat{\mathbf{F}}^l$ by a factor of two using transposed convolutions [29]. It introduces its bottleneck block [11] to generate new feature maps based on the concatenation of the upsampled maps and the higher-resolution raw feature maps $\mathbf{F}^{l-1}$. The following two bottleneck cells repeat this process as shown in Fig.3, and the channel numbers of their generated feature maps are the same as those of the corresponding upsampled feature maps. The standard convolutional layer takes the outputs of the final bottleneck cell and produces a one-channel feature map $\mathbf{X}$ at a coarse scale such as 56×56. Finally, for this coarse mask $\mathbf{X}$=$\mathcal{G}(\mathcal{E}(\mathbf{I}),t)$, the upsampling based on bilinear interpolation is employed to obtain a smoother mask at the image scale as $\mathbf{M}$=$\text{upsample}(\mathbf{X})$.

Now we consider the optimization of the spatial filter and the remaining parts of the mask generator. Similar to metric learning [14], the class-related embedding vectors in the spatial filter can be gradually updated by maximizing the similarity to the feature vectors of the same-class images while minimizing the similarity to those of different-class images. Thus, we assign training images with true labels (k=t) and fake labels (k≠t) iteratively and update the embedding vectors $\mathbf{V}_k$ by minimizing the following loss:

$$\ell_{\text{embed}}(\mathbf{F}^l,\mathbf{V}_k) = \begin{cases} -\sum(\text{sigmoid}({\mathbf{F}^l_{ij}}^{\text{T}}\mathbf{V}_k)), & k=t; \\ \sum(\text{sigmoid}({\mathbf{F}^l_{ij}}^{\text{T}}\mathbf{V}_k)), & k\neq t. \end{cases} \tag{4}$$

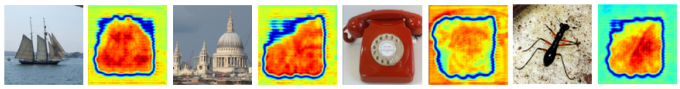


Fig. 4. The examples of the masks obtained with non-monotonic mappings, where higher scores can not guarantee larger contributions.

After that, the remaining parts of the mask generator can be optimized based on its generated mask $\mathbf{M}$ via Eq.2. Since the mask generator is trained offline, we can obtain a real-time explanation based on a single forward-pass.

3.2.3 *Limitation Analysis*

Lack of differentiation. The perturbation-based explanations are formulated to distinguish the supporting features from the background and are generally optimized based on a large number of iterations. Although mask generators significantly accelerate the explanations, they still fail to consider the different degrees of importance of the supporting features, and their obtained masks are lack of differentiation.

Sensitive hyper-parameter tuning. During the training phase, balancing the trade-offs between the classification loss and the additional soft constraints, e.g., the smoothing term in Eq.2, involves a non-trivial hyper-parameter tuning process. Since the quality of masks is subjective for evaluation, it increases the burden of learning a good generator where the Bayesian optimization is hard to employ [28].

3.3 Principles of Controllers

To produce differentiated saliency maps, we first introduce the concept of distribution controllers $\mathcal{C}$, which guides the relevance scores towards desired distributions. We place the controller right after the generator to together compose the differentiated relevance estimator DRE. Suppose $\mathbf{X}$ is the initial output of the generator (as we mentioned in Sec. 3.2.2), and $\mathbf{Z}$ denotes the output of $\mathcal{C}$. The output of the distribution controller is expressed as

$$\mathbf{Z} = \mathcal{C}(\mathbf{X}). \tag{5}$$

Besides, we follow [4], [6] to upsample $\mathbf{Z}$ with interpolation, aiming to improve the smoothness at the image scale. An overview of our framework is provided in Fig.3.

We investigate the principles for the controller design.

Principle 1. The hyper-parameters in $\mathcal{C}$ can be *easily set* without prior knowledge of classifiers and datasets.

Principle 2. The output relevance scores of $\mathcal{C}$ approach a *right-skewed distribution* over (0,1) for each decision.

Principle 3. The mapping function from the distribution controller's input $\mathbf{X}$ to its output $\mathbf{Z}$ is *monotonic*.

For the illustration of the last two principles, two examples are shown in Fig.2 and Fig.4, respectively. In the first figure, by modifying the expected distributions of the outputs of $\mathcal{C}$ from the right-skewed distribution to the left-skewed distribution, the differentiation of the saliency map is remarkably reduced. Besides, the proportions of pixels highlighted with high scores are positively correlated with the area at the right part of pre-configured distributions. It is worthwhile noting that the quantitative observation in [2], [7] shows that the occlusion of 5% (25%) pixels in natural images can bring nearly 50% (90%) drop in classification

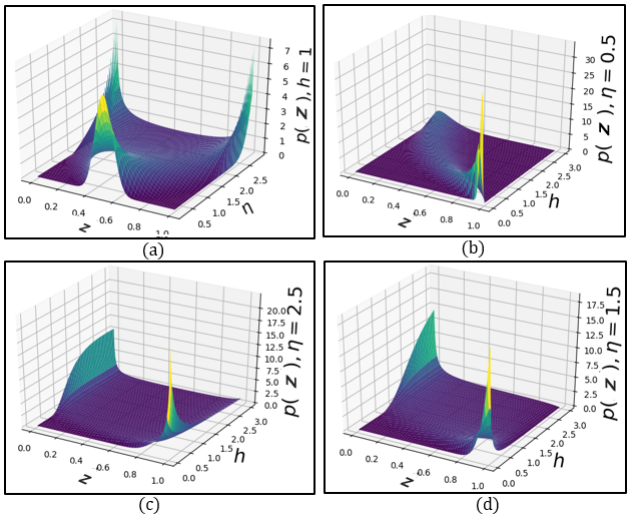


Fig. 5. The PDFs with the different settings of hyper-parameters.

confidence, which again demonstrates the effectiveness of the right-skewed distributions. In the second figure, a non-monotonic transform $g(x)$=x^2 is used in $\mathcal{C}$, making the scores deviate from the expected meaning. That is, a higher score implies a larger contribution. In contrast, a monotonic mapping enables to enhance the differentiation of a saliency map without changing the ranking of its relevance scores.

3.4 Controller Design

Following the above principles, we introduce a simple design of the controller. Since the sum of two independent random variables is more normal (Gaussian) than the original variables [12], [13], we first assume that the distribution of the inputs of the controller $\mathcal{C}$ (the outputs of the generator $\mathcal{G}$) is nearly normal for convenience. Later we show that the proposed controller built upon this assumption also guides other distributions towards the right-skewed ones.

3.4.1 *From the Normal to the Standard Normal*

To obtain the desired distributions, we first introduce instance normalization [26] to guide the normal distribution towards the standard normal distribution. The goal of this step is to shift the scores around the opposite sides of zero.

Specifically, let $x_{ij}\in\mathbf{X}$ be the input entry at the location (i,j), and $y_{ij}\in\mathbf{Y}$ denotes the expected variable following a standard normal distribution. The mapping can be expressed as

$$y_{ij} = \omega(x)_{ij} = (x_{ij} - \mathbb{E}[x_{ij}])/(\sqrt{\mathbb{V}[x_{ij}]}), \tag{6}$$

where the expectation $\mathbb{E}[\cdot]$ and variance $\mathbb{V}[\cdot]$ are computed over the entries of each $\mathbf{X}$.

3.4.2 *From the Standard Normal to the Right-skewed*

Now we guide the above scores towards right-skewed distributions monotonically. To do this, we introduce a customized transformation function with easy-to-set hyper-parameters.

To produce the relevance scores with a right tail in (0,1), we first transform the normal distribution towards a uniform distribution based on sigmoid functions and then change the skewness of the distribution based on power functions. An illustrative example is shown in the bottom-left in Fig.3. Specifically, the output z_{ij} of $\mathcal{C}$ is obtained as

$$z_{ij} = \varphi(y_{ij}) = (\mathrm{sigmoid}(\eta \cdot y_{ij}))^h = (\frac{1}{1+\mathrm{e}^{-\eta\cdot y_{ij}}})^h \tag{7}$$

where η aims to guide the new scores approach the uniform distribution [27], and h determines the skewness of the final distribution.

In particular, the hyper-parameters η and h can be easily set according to their effects on the transformed probability density function (PDF) $p(z)$. To do this, we introduce the probability density transformation [9] and obtain $p(z)$ as

$$p(z) = \frac{1}{\sqrt{2\pi}h\eta}\cdot\frac{1}{z(1-z^{1/h})}\cdot \mathrm{e}^{-\frac{(\ln(z^{(-1/h)}-1))^2}{2\eta^2}}, \tag{8}$$

The detailed proof can be found in Appendix A.

Analysis. Now we set the hyper-parameters based on their effects on the intuitive geometry of $p(z)$, which corresponds to the distribution of relevance scores.

Firstly, we fix h=1 and observe the effect of η. The corresponding PDFs are displayed in Fig.5(a). By changing η within (0.5,2.5), $p(z)$ remains its skewness and changes from the concave to the convex for $z\in$(0,1). In particular, η=1.5 approximately leads $p(z)$ to an uniform distribution. Considering 1.5≈0.9$\sqrt{\pi}$, it is consistent to the sigmoid approximation of the cumulative probabilities of the standard normal distribution [27].

Secondly, we set η=$\{$0.5,1.5,2.5$\}$ and observe the effect of h. Three PDF figures are displayed in Figs.5(b-d), where all $p(z)$s are able to obtain right-skewed distributions under a large h. However, the geometries of these $p(z)$s are significantly different. Fig.5(b) shows that $p(z)$ with η=0.5 obtains extremely low probabilities for $z\in$(0.5,1). Once a relevance score larger than 0.5 appears and becomes an outlier, this range for highlighting strong supporting features is likely to be wasted. Fig.5(c) shows that $p(z)$ with η=2.5 continues the undesired convexity and leads to more high scores than middle scores. $p(z)$ with η=1.5 can lead to a clear tail over the range of (0,1), as shown in Fig.5(d).

Above all, η=1.5 enables the sigmoid approximation of the cumulative probability for the standard normal distribution and leads it to a uniform distribution [27]. With h=2.5, we can further obtain the scores under the right-skewed distribution with a clear tail over (0,1). Note that other transformations with monotonicity can also be considered.

3.4.3 *Effects on Other Distributions.*

Now we relax the normal distribution on $\mathbf{X}$ and analyze the effects of the above controller (η=1.5 and h=2.5) on other typical distributions, including uniform distributions, the mixtures of normal distributions, and skew normal distributions. For simplicity, we only show the results based on the synthetic data and leave the detailed analysis in Appendix B. The original distributions (the 1st row) and their transformed distributions (the 2nd row) are displayed in Fig.6. As we can see, although the controller may not transform them into the right-skewed distributions completely, it still shifts a majority of relevance scores towards lower values and remains a minority of the scores at high values, which makes the scores away from the left-skewed distributions.

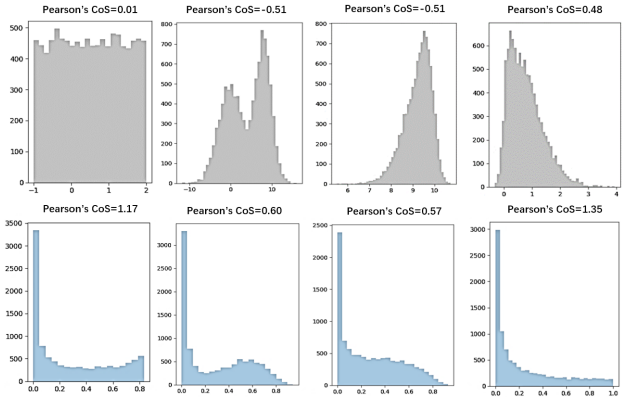


Fig. 6. The effects of the controller on the synthetic data with different distributions. The 1st and the 2nd rows show original distributions and their transformed distributions, respectively. For a quantitative comparison, we also list their Pearson's Coefficients of Skewness (CoS) [5].

From a quantitative perspective, we also calculate their Pearson's Coefficients of Skewness (CoS) [5] for comparison. Specifically, if skewness is positive (negative), the relevance scores are right- (left-) skewed, meaning that the right (left) tail of the distribution is longer than the left (right). As we observe, our controller consistently increases the values of the skewness for the above typical distributions.

3.5 Estimator Optimization

This section pays attention to the optimization of the above differentiated relevance estimator DRE. Denote ℓ_{class} as the classification loss used for training DNNs, such as the commonly used cross-entropy loss. Considering the explanation is already guided to be differentiated, DRE can be directly optimized without the non-trivial hyper-parameter tuning:

$$\text{argmin}_{\mathcal{G}} \ell_{\text{class}}(f(\psi(\mathbf{I}, \mathbf{M})), t), \text{ where } \mathbf{M} = \text{upsample}(\mathcal{C}(\mathbf{X})). \tag{9}$$

In particular, Eq.9 enables us to improve the faithfulness of explanations, since it is simplified to find the region that maximizes the target probability under the expected distribution. It is also valuable to train the classifier and the relevance estimator at the same time. As this paper focuses on post hoc explanations, we leave it for our future work.

4 Experiments

This section investigates the performance of DRE. We first evaluate the faithfulness and the explainability based on object recognition and scene recognition tasks. Then, we introduce simple tricks to further improve the performance. Finally, we perform targeted ablation studies to empirically demonstrate the effectiveness of the proposed innovations and discuss the results for misclassifications.

4.1 Setup

To demonstrate the broad applicability, we apply the proposed DRE to 3 types of CNNs, including ResNet50 [11], VGG19 [21], and GoogleNet [25]. The following 9 methods are used for comparison: (1) Mask Generator (MGnet) [4], (2) Meaningful Perturbation (MPert) [8], (3) Grad CAM (GCAM) [19], (4) Grad CAM++ (GCAM++) [2], (5) Feature Inversion (FInv) [6], (6) Extreme Perturbation (XPert) [7], (7) Vanilla Gradient (VGrad) [20], (8) Smoothness Gradient (SMGrad) [22], (9) Integrated Gradient (ITGrad) [24]. Of note, most of them require a large number of forward-and-backward iterations to build each mask and involve sensitive hyper-parameters in their objective functions. A summary is shown in Tab.2. We do not regard easy-to-set hyper-parameters as sensitive ones, such as the number of iterations[2] in MPert. We empirically tune the sensitive hyper-parameters around their suggested values.

TABLE 2
Characteristics of compared methods. The easy-to-set hyper-parameters are not regraded as sensitive ones.

	number of sensitive hyper-parameters	number of iterations per explanation
DRE	-	1
MGnet	3	1
MPert	2	300
FInv	3	80
XPert	-	300
GCAM	-	1
GCAM++	-	1
VGrad	-	1
SMGrad	1	50
ITGrad	-	200

Implementation. For each of the above CNNs, we divide its convolutional layers into a few groups based on the resolutions of their outputs. We then introduce the three bottleneck cells at the intermediate positions to get the raw feature maps. We use varying channel numbers for different CNNs, aiming to propagate sufficient information between the cells while keeping efficiency. Specifically, for ResNet50 which consists of the intermediate layers with {256,512,1024} channels, we introduce one quarter channels for the high-to-low-resolution cells, namely {64,128,256}; for VGG19 and GoogleNet that contain the intermediate layers with {128,256,512} and {192,480,832} channels, we half the channel numbers for the corresponding cells, namely {64,128,256} and {96,240,416}, respectively. We use a two-stage scheme to train the relevance estimator. We first train class-related spatial filters based on the sampled images from the training set and then optimize other parts of the relevance estimator for 10 epochs. Of note, no ground truth is introduced, and only the outputs of the classifiers are utilized. We set the batch size to 64 and use Adam with the initialized learning rate of 10^{-2}. We apply the step decay and reduce the learning rate by half every three epochs. During the second stage, 50% background images are set to the Gaussian blurred version of raw images with the variance of 10, and the remaining ones are set to random color images with the addition of Gaussian noise. For a fair comparison, all perturbation-based methods apply the same strategy for adding perturbations.

Quantitative metrics. The faithfulness and the explainability of saliency maps are supposed to be evaluated based on the relevance scores of all pixels. Here we introduce two generalized metrics based on the ranking of pixels.

2. A larger iteration number empirically brings in better performance.

TABLE 3
Ranking-based quantitative evaluation on faithfulness $\mathcal{M}_{\mathcal{F}}$.

	ImageNet				Birds-200-2011			
	ResNet50	VGG19	GoogleNet	MEAN	ResNet50	VGG19	GoogleNet	MEAN
DRE	77.13	76.75	69.24	74.37	84.62	85.25	82.13	84.00
MGnet	56.61	60.09	50.57	55.75	64.97	75.90	75.32	72.06
MPert	72.40	73.92	64.00	70.10	76.08	82.31	75.09	77.83
FInv	66.73	67.31	61.50	65.18	75.14	78.53	78.52	77.40
XPert	62.07	67.55	52.98	60.87	76.90	80.82	75.84	77.85
GCAM	70.13	64.56	67.84	67.51	77.67	80.49	79.43	79.20
GCAM++	68.34	69.96	62.51	66.94	78.42	79.75	78.34	78.84
VGrad	18.80	13.57	15.43	15.93	12.23	11.51	10.12	11.29
SMGrad	29.38	35.47	40.26	35.03	46.97	48.80	56.58	50.78
ITGrad	16.47	20.35	43.37	26.73	20.71	22.03	35.19	25.98

TABLE 4
Ranking-based quantitative evaluation on explainability $\mathcal{M}_{\mathcal{E}}$.

	ImageNet			
	ResNet50	VGG19	GoogleNet	MEAN
DRE	83.02	83.91	81.73	82.89
MGnet	82.62	83.53	81.87	82.67
MPert	75.74	72.50	70.98	73.07
FInv	75.20	72.63	75.35	74.39
XPert	78.73	80.92	71.26	76.97
GCAM	79.28	74.93	82.15	78.79
GCAM++	82.86	84.46	83.20	83.51
VGrad	66.23	70.90	66.78	67.97
SMGrad	73.17	74.03	70.31	72.50
ITGrad	66.92	66.74	63.25	65.64

Faithfulness. The traditional metrics use heuristic segmentation strategies on the scores of each mask and calculate the probability of the target class based on a fixed ratio of clean high-score pixels, which reduces the fairness for comparing different methods [8]. We instead utilize the ranking of pixels and perform the evaluation based on the target class probabilities corresponding to a number of ratios. Suppose Δ is the interval for the ratio of clean high-score pixels, and $\mathcal{S}_i$ denotes the set of locations with the top $i\times\Delta$ highest scores. For each image, we first estimate the probability of its fully blurred version as $Q_0{=}f_t(\psi(\mathbf{I},\mathbf{0}))$. Then we replace its blurred pixels within $\mathcal{S}_i$ by the corresponding $i\times\Delta$ pixels in the clean image and estimate the probability of the new image as Q_i. We repeat this step by increasing the ratio of clean pixels, until reaching the fully clean image and obtaining $Q_m{=}f_t(\psi(\mathbf{I},\mathbf{1}))$ ($m\times\Delta{=}1$). With the intervals of Δ, the area under the curve (AUC) of the probability vs. the ratio is used as the measure of faithfulness:

$$\mathcal{M}_{\mathcal{F}} = 100\% \times \sum_i Q_i \cdot \Delta,\ i = 1, 2, \ldots, m. \tag{10}$$

Explainability. The explanation with high explainability should provide clear reasons that are easy to understand. Since it is time-consuming for users to detect the locations of all meaningful features (including bias features), we generally use bounding boxes as an alternative for its evaluation [6], [8]. For example, weakly-supervised object localization evaluates masks by calculating the intersection over the union between their binary variants and bounding boxes. Nevertheless, it faces the issue of choosing thresholds. Thus, we introduce a new metric by regarding the relevance scores as the results of retrieval tasks [1]. Specifically, let $\mathcal{S}_i$ be the set of locations that obtain the top i highest relevance scores, and $\mathcal{S}_b$ indicates the set of locations within the bounding box. We calculate the precision $P_i{=}\frac{|\mathcal{S}_b\cap\mathcal{S}_i|}{|\mathcal{S}_i|}$ as the fraction of these i locations retrieved within bounding boxes, and the recall $R_i{=}\frac{|\mathcal{S}_b\cap\mathcal{S}_i|}{|\mathcal{S}_b|}$ as the fraction of the within-bounding-box locations that are retrieved within these i locations. By computing P_i and R_i for all $\mathcal{S}_i$s, we can evaluate the explainability by the AUC of the precision vs. the recall as

$$\mathcal{M}_{\mathcal{E}} = 100\% \times \sum_i P_i \cdot (R_i - R_{i-1}). \tag{11}$$

4.2 On Object Recognition

This section investigates the effectiveness of the proposed method in object recognition tasks, which is the primary motivation of introducing right-skewed distributions. We use two real-world image datasets ImageNet and Birds-200-2011 for evaluation, where the latter is a fine-grained dataset of 200 bird species. In particular, we load pre-trained CNNs from torchvision for ImageNet and train ResNet50, VGG19, and GoogleNet to build the classifiers for Birds-200-2011.

4.2.1 Ranking-based Quantitative Evaluation

On the faithfulness with $\mathcal{M}_{\mathcal{F}}$. To evaluate the faithfulness of the proposed relevance estimator, we calculate the mean of the metric $\mathcal{M}_{\mathcal{F}}$ based on 10,000 and 2,000 sampled images for ImageNet and Birds-200-2011, respectively. We set $\Delta{=}\frac{1}{32}$ as the interval. Besides, we add a smoothed mask over the original one with a small weight. We introduce min-max normalization on $\mathcal{M}_{\mathcal{F}}$s of all methods for each image, balancing the effects of different images.

The results of the average $\mathcal{M}_{\mathcal{F}}$s are displayed in Tab.3, where the following observations can be obtained. *Firstly*, VGrad, SMGrad, and ITGrad are generally worse than the others with a large gap. It is understandable that, these methods search sensitive pixels based on the gradients, and the pixels with high scores will be discretely distributed in each image. As a result, it becomes harder for them to gather sufficient supporting information in a local receptive field and recover a high class probability. *Secondly*, GCAM obtains comparable performance to GCAM++ on average, and MPert that directly optimizes masks in a high resolution also brings satisfying faithfulness. *Thirdly*, DRE outperforms all the other methods and enjoys much better performance

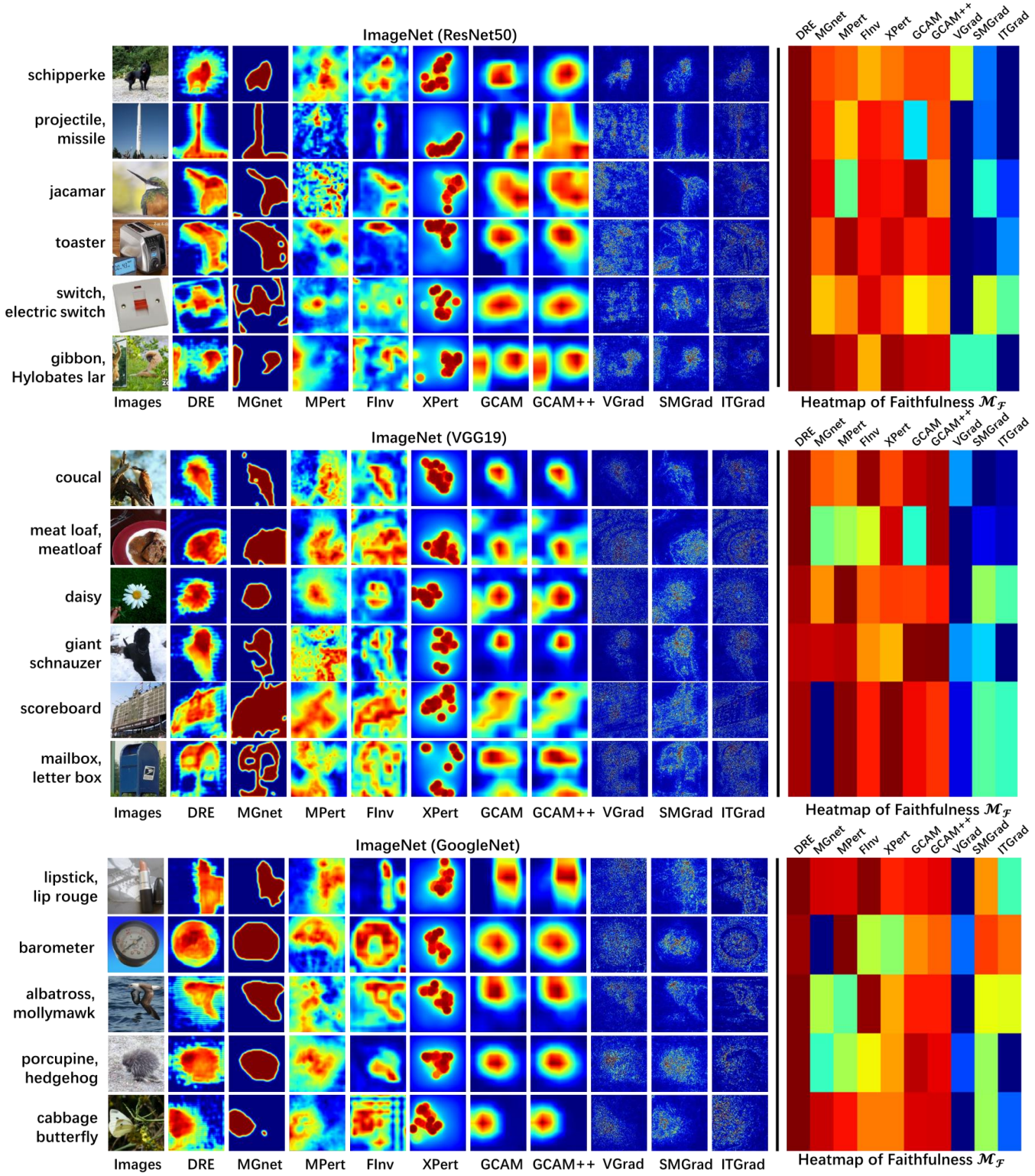


Fig. 7. The saliency maps of different explanation methods for the CNNs trained on ImageNet, in which these sampled images obtain correct classifications. We also display the heatmap of their normalized $\mathcal{M}_{\mathcal{F}}$ values.

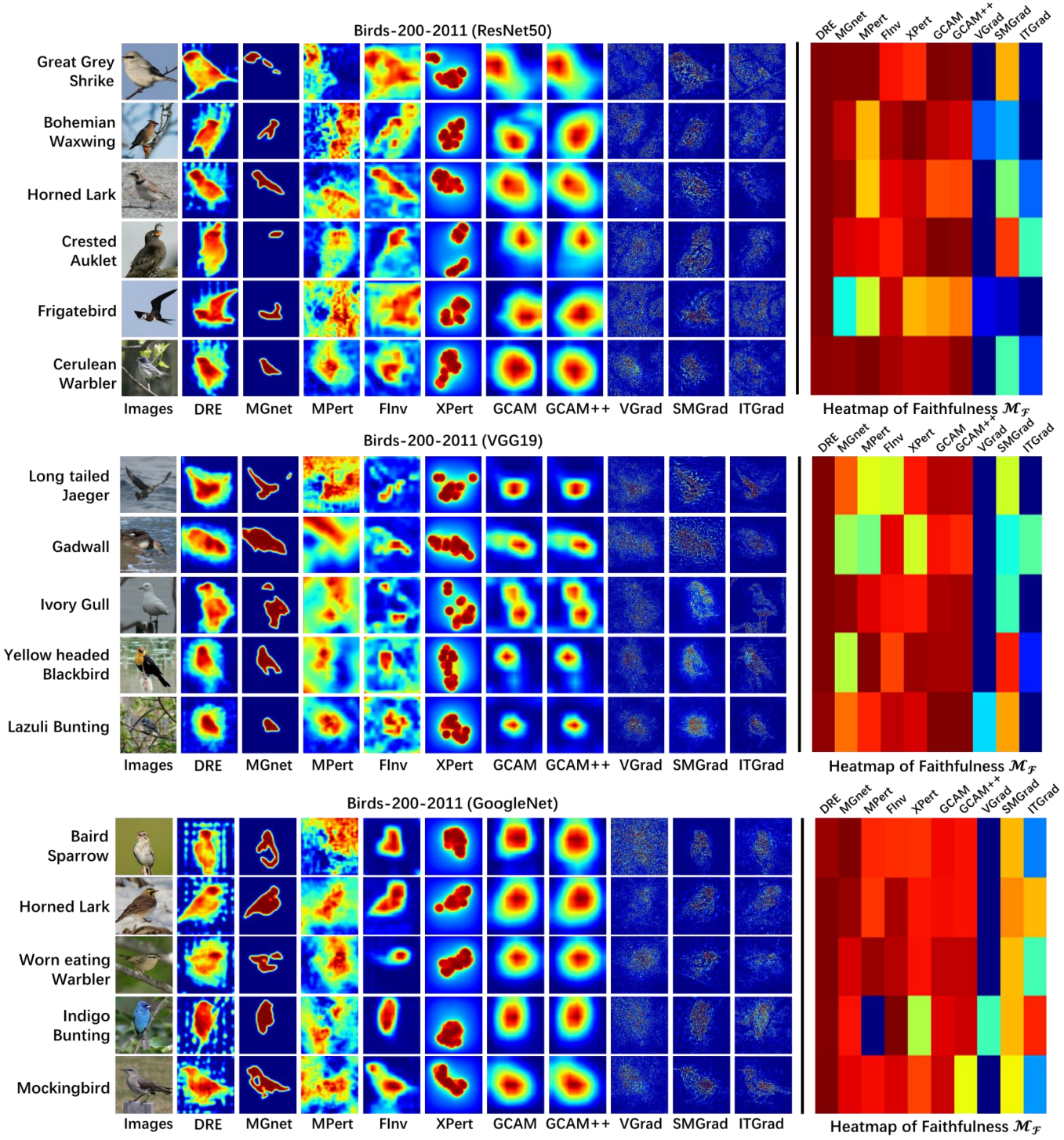


Fig. 8. The saliency maps of different explanation methods for the CNNs trained on Birds-200-2011, in which these sampled images obtain correct classifications. We also display the heatmap of their normalized $\mathcal{M}_{\mathcal{F}}$ values.

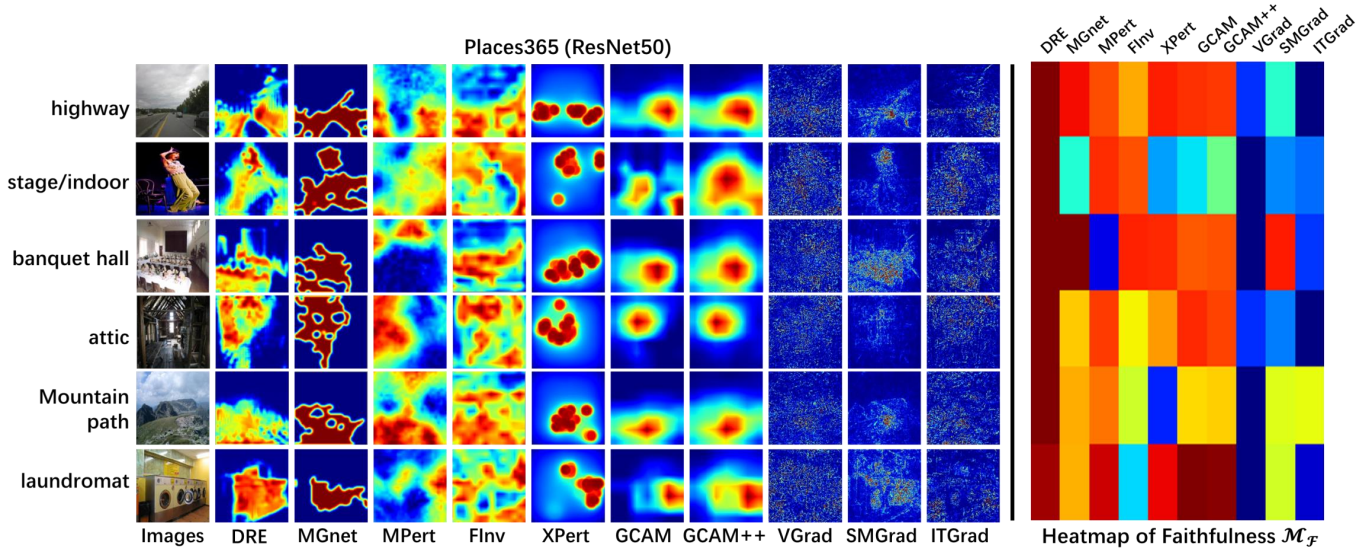


Fig. 9. The saliency maps of different explanation methods for the ResNet50 trained on Places365, in which these sampled images obtain correct classifications. We also display the heatmap of their normalized $\mathcal{M}_{\mathcal{F}}$ values.

than MGnet. Since the two methods apply the same architectures for their mask generators, the results demonstrate that guiding the relevance scores towards right-skewed distributions improves the ranking of supporting features.

On the explainability with $\mathcal{M}_{\mathcal{E}}$. To reveal the explainability, we introduce $\mathcal{M}_{\mathcal{E}}$ by evaluating the performance of object localization. For this, 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 10,000 validation images of ImageNet with bounding box annotations.

The results of average $\mathcal{M}_{\mathcal{E}}$s are listed in Tab.4, where we obtain the following observations. *Firstly*, the last three gradient-based methods generally perform worse than the others. The possible reason is that, gradients are insensitive to the smooth supporting regions, which makes these regions ignored and harms the ranking of pixels. *Secondly*, GCAM++ obtains better performance than GCAM and the other methods. Understandably, the former is designed to detect multiple objects in the image and assign them high relevance scores. *Finally*, by replacing all constraints with a simple distribution controller, DRE outperforms MGnet with a small gap, and both of them enjoy better performance than most other methods. The reason is that, benefiting from the training with large-scale images, they tend to generate high relevance scores for the supporting features that are robust to the target class. If the bias features do not play the main role in the classification, the corresponding supporting regions prefer the locations inside the objects.

To sum up, DRE obtains comparable or even better explainability to others but achieves much higher faithfulness. Therefore, although we are not able to analyze the effects of bias features without more human intervention, the synthetic results still demonstrate its effectiveness empirically.

4.2.2 Visualization-based Qualitative Comparison

Below we visually compare different explanation methods based on their obtained saliency maps. The red and blue colors denote the high and low scores, respectively. We sample images from ImageNet and Birds-200-2011, and show their results in Fig.7 and Fig.8. In particular, we also display their normalized $\mathcal{M}_{\mathcal{F}}$ values via a heatmap, in which the color of each rectangle represents the faithfulness of the saliency map at the corresponding location.

From these results, we have the following observations. *Firstly*, the gradient-based methods bring more low relevance scores for the pixels insides the objects and high relevance scores for the pixels outside the objects. Considering their lower faithfulness, this observation implies that the supporting pixels generally locate inside the objects as expected, demonstrating the effectiveness of estimating the explainability with bounding boxes. *Secondly*, since GCAM and GCAM++ only take high-level feature maps into account, they fail to provide a detailed estimation of relevance scores. This issue becomes more obvious for fine-grained images, where they can only detect the locations of the objects. Besides, GCAM and GCAM++ may still miss a majority of supporting pixels of objects, such as the 4th row in Fig.8. *Thirdly*, MGnet is case-sensitive, which will either detect all supporting pixels or a few most important ones. Although we can carefully adjust the hyper-parameters for each kind of datasets and networks, the mask generators need to be re-trained for each setting, reducing its practicability significantly. Similarly, MPert and FInv contain sensitive hyper-parameters that need to be tuned for each instance individually. *Fourthly*, MGnet assigns the detected pixels similar relevance scores and loses the differentiation. We have attempted to perform the controller on its obtained masks in a post hoc manner, which still fails to build differentiated masks. The reason is that its relevance scores are already stacked at 0 and 1. *Finally*, DRE not only obtains high scores for the supporting pixels inside the objects and brings higher faithfulness, but also displays the different degrees of importance of these pixels, e.g., the supporting pixels within the animals' heads are more important than those within the other parts.

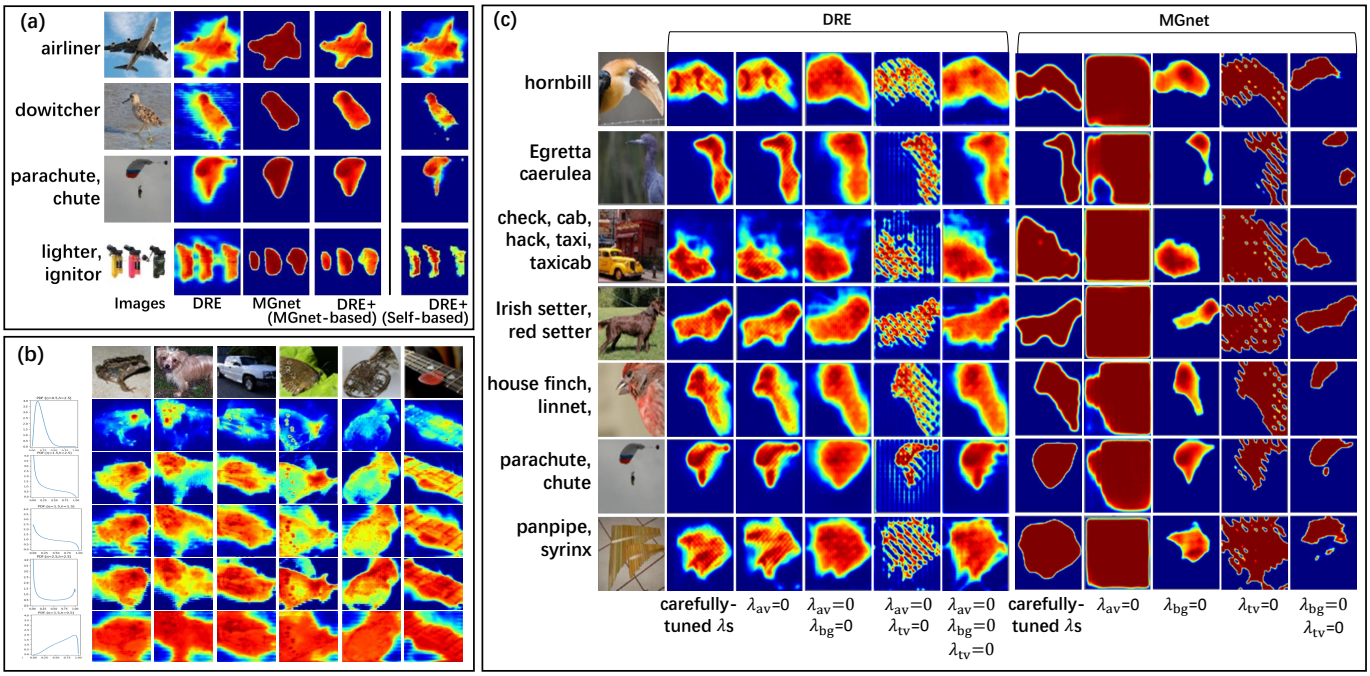


Fig. 10. The illustrative examples for Sec. 4.4, which are built upon ResNet50 with ImageNet. (a) The effects of post hoc tuning tricks on saliency maps. (b) The effects of different pre-configured distributions (the 1st column) on saliency maps (the 2nd-7th columns). Each row of masks corresponds to a pre-configured distribution. (c) The effects of the ad hoc constraints on DRE and MGnet [4].

TABLE 5
Ranking-based quantitative evaluation on faithfulness $\mathcal{M}_{\mathcal{F}}$.

Places365 (ResNet50)				
DRE	MGnet	MPert	FInv	XPert
70.45	53.85	64.61	64.84	55.71
GCAM	GCAM++	VGrad	SMGrad	ITGrad
68.60	60.89	7.73	38.07	45.20

4.3 On Scene Recognition

Now we introduce the explanation methods for the CNNs trained on scene images. Specifically, we first train ResNet50 on Places365 and then regard it as the black-box classifier. We list their faithfulness metric $\mathcal{M}_{\mathcal{F}}$ in Tab.5 and show some of their obtained saliency maps in Fig.9.

From these results, we obtain the following observations. *Firstly*, comparing to the masks for object images, the high scores of DRE and MGnet for scene images may locate at more than one region of an image. Understandably, scenes are composed of objects, and the conception of scenes is more comprehensive than that of objects. *Secondly*, although the masks of DRE tend to be visually noisy compared with GCAM and GCAM++, they still detect the important regions clearly and lead to high faithfulness. From the quantitative perspective, we also observe that DRE obtains higher faithfulness than all the other methods.

4.4 Discussion

In this section, we first introduce some simple tricks to further improve the proposed method. After that, since the right-skewed distributions are used and the ad hoc constraints are ignored, their impacts are investigated via targeted ablation studies. Finally, we discuss the saliency maps corresponding to misclassifications.

4.4.1 On Improvements with Simple Tricks

Although DRE has achieved stratifying performance without the non-trivial hyper-parameter tuning, constraining the scores of various images towards the same pre-configured distribution may lead to low but redundant scores on backgrounds, especially when the supporting pixels only take a tiny part of all pixels. Thus, we present two simple tricks to further improve the differentiation of DRE, including one self-based and one MGnet-based. For convenience, only ResNet50 is used as the classifier.

DRE+ (Self-based). This trick improves DRE based on the saliency map itself. Given an estimated mask with the relevance scores of pixels, we first follow the process in the faithfulness metric to iteratively calculate the probability Q_i based on the top $i\times\Delta$ pixels. Next, we modify these probabilities with $Q_i{=}\max Q_{j\leq i}$ to obtain the monotonicity. For efficiency, we only calculate the probability Q_i at the $i\times\Delta$-th pixels and infer the probabilities at the remaining pixels with linear interpolation. Since the k-th pixel is not likely to be the supporting one if Q_k already approaches the maximum $Q_{\max}$ ($\max_k Q_k$), we perform the post hoc tuning on the relevance scores as $M'_k{=}M_k{\times}W_k$, where $W_k{=}(Q_{\max}{-}Q_k)$, and M_k is the original relevance score.

DRE+ (MGnet-based). It is natural to cooperate the proposed DRE with MGnet [4], since the latter can obtain clearer boundaries between objects and backgrounds. Thus, we introduce another mask as $\mathbf{M}' = \mathbf{M}_{\mathrm{DRE}} \odot \mathbf{M}_{\mathrm{MGnet}}$. For simplicity, we perform the post hoc combination without training them together with the shared mask generator.

The obtained masks showing the effects of DRE+ (Self-based) and DRE+ (MGnet-based) are displayed in Fig.10(a),

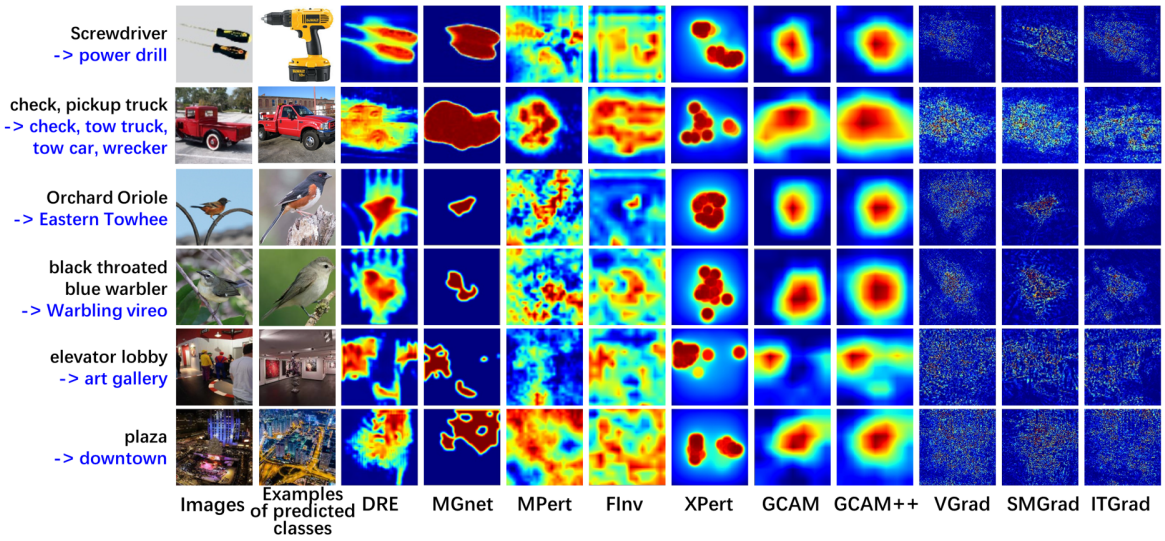


Fig. 11. The examples of the masks of different methods on ImageNet (1st-2nd rows), Birds-200-2011 (3rd-4th rows), and Places365 (5th-6th rows), where ResNet50 is used as the classifier. Of note, labels in black indicate the ground truths, and those in blue denote the predicted classes.

where the masks of original DRE and MGnet are also provided. From the results, the following observations can be made. *Firstly*, compared with MGnet, DRE without post hoc tuning sometimes sacrifices the boundaries between objects and backgrounds. *Secondly*, by combing the masks of DRE and MGnet, DRE+ (MGnet-based) improves the boundaries while keeping the differentiation to some degree. *Thirdly*, owing to the better ranking of the scores over all pixels (demonstrated by $\mathcal{M}_{\mathcal{F}}$ in Sec. 4.2.1), DRE+ (Self-based) can effectively utilize DRE's masks to reduce the noise around the boundaries and highlight the supporting pixels with differentiated scores. Since it does not introduce extra hyper-parameters, DRE+ (Self-based) is more practical. We further introduce flexible variants of self-based tuning in Appendix C, which could be more beneficial to DRE.

4.4.2 On Distribution Controller: the Effects of Distributions.

To show the effectiveness of the proposed controller, we first change the setting of its hyper-parameters, which leads the input with normal distributions to different types of pre-configured distributions. Specifically, we set (η,h) to (0.5,2.5), (1.5,2.5), (1.5,1.5), (2.5,2.5), (1.5,0.5) and the 5 corresponding distributions are shown in the 1st column in Fig.10(b). For convenience, we only use ResNet50 as the classifier and train the relevance estimators for 3 epochs. The examples of the corresponding saliency maps are shown in the following columns in the same figure.

As we can see, although it is hard to expect that all masks obtain relevance scores with the same distributions as the pre-configured ones, the controllers consistently enforce them towards these distributions. In general, the estimators built upon the right-skewed controllers obtain the differentiated masks, and the estimators built upon the left-skewed controllers reduce the explainability significantly.

4.4.3 On Optimization: the Effects of Ad Hoc Constraints.

To evaluate the impact of our simplified objective function, we introduce an ablation study to analyze how the ad hoc constraints affect the proposed relevance estimator. Following the formulation of MGnet with Eq.2, we add them back to Eq.9 and train the estimator using the following hyper-parameter settings: (1) carefully-tuned λs, (2) λ_{av}=0, (3) λ_{av}=λ_{bg}=0, (4) λ_{av}=λ_{tv}=0, (5) λ_{av}=λ_{bg}=λ_{tv}=0. For comparison, an ablation study is also performed on MGnet with the setting of (1) carefully-tuned λs, (2) λ_{av}=0, (3) λ_{bg}=0, (4) λ_{tv}=0, (5) λ_{bg}=λ_{tv}=0. For efficiency, we only use ResNet50 as the black-box classifier and train the corresponding estimators for 3 epochs. The masks corresponding to various settings are displayed in Fig.10(c).

From the results, the following observations can be obtained. *Firstly*, comparing the 6th column with the 2nd column, we can see that by adding all the constraints back, the masks of DRE can be improved to some degree. However, we also observe from the 3rd column that λ_{av}=0 has few effects on our relevance estimator. Besides, λ_{bg}=0 will increase the noise around the boundaries owing to the smoothness constraint (in the 4th column), and λ_{tv}=0 causes the holes in the masks (in the 5th column). Nevertheless, by further removing these two terms, these shortcomings can be alleviated, and the final masks of DRE remain the satisfying quality (in the 6th column). *Secondly*, all the constraints in MGnet, however, result in remarkable impacts on its masks, especially λ_{av}=0 (in the 8th column). Although λ_{bg}=0 can relax the masks and improve their differentiation, it would ignore a part of supporting features, such as the 4th-5th rows in the 9th column. Besides, it enhances the sensitiveness of λ_{av}. For example, while λ_{bg} is carefully set, λ_{av} varying within (2,12) consistently leads to acceptable results for ResNet50. Once λ_{bg}=0, the quality of masks is acceptable only for $\lambda_{\text{av}}\in$(2,4). The reason is that, minimizing $f_t(\psi(\mathbf{I},\mathbf{1}\text{-}\mathbf{M}))$ can avoid the supporting pixels being regarded as backgrounds. Once this term is removed, the size of the supporting pixels is totally controlled by the hyper-parameter λ_{av}. When it is slightly larger, a part of supporting pixels will be regarded as backgrounds. In short, benefiting from the distribution controllers, DRE can

be insensitive to the ad hoc constraints.

4.4.4 On Misclassifications

As the objective of explanation methods aims to explain all decisions, the explanations on misclassifications also need to be investigated. We thus show the masks of different methods on misclassified images. We take ResNet50 as an example and display the masks of the images from different datasets in Fig.11. As we can see, most explanation methods still target the typical parts of objects or the representative objects of scenes. It implies that different classes may share the same visual features, which leads to misclassifications. For example, the object images in the 1st-2nd rows share the similar shapes with the predicted classes, the bird images in the 3rd-4th rows share the similar colors with the wrong classes, and the scene images in the 5th-6th rows focus on the similar objects to the predicted classes. This observation is in line with the finding in [30], where different classes of scene images share the same object-level features.

5 Conclusion and Future Work

In this paper, we introduce a simple but effective relevance estimator called DRE to provide differentiated explanations for the decisions of DNNs. Specifically, we present the concept of distribution controllers on relevance scores and integrate it with a trainable mask generator to directly guide the relevance scores. By analyzing the effects of the skewness of the pre-configured distributions, we develop a simple distribution controller with the right-skewed distribution. We optimize DRE under the classification loss without non-trivial hyper-parameter tuning, which also improves the faithfulness of explanations. For each of the above innovations, we perform the targeted experiments to investigate their effectiveness. Finally, we compare DRE with state-of-the-art methods, and the experimental results demonstrate that DRE significantly improves faithfulness with high explainability.

There are some aspects needing further investigations. Firstly, although this paper provides an intuitive comparison of the transformed distributions for setting hyper-parameters, a quantitative analysis of the ratio of features at the tail is preferable. Secondly, since self-based tuning can improve saliency maps, it is worth incorporating it into the original models for an end-to-end optimization. Thirdly, benefiting from the simpleness of using distribution controllers, explaining the decisions of graph neural networks based on the controllers becomes another possible direction.

Appendix A
The Proof of Eq.8 in Sec.3.4.2

Let $p(y)$ denote the probability density function (PDF) of the variable y, and the transformed variable z is calculated as $z = \varphi(y)$. According to the probability density transformation [9], the transformed PDF $p(z)$ can be obtained as

$$p(z) = p_y(\varphi_y^{-1}(z)) \cdot |\frac{\partial \varphi_y^{-1}(z)}{\partial z}|, \tag{12}$$

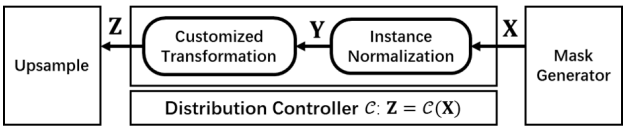


Fig. 12. The data flow inside the distribution controller with variables $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$.

where $\varphi_y^{-1}(z)$ denotes the inverse function of z on y, and $p_y(\varphi_y^{-1}(z))$ means substituting the above result into the PDF of y. Based on Eq.7 in the paper, we obtain

$$y = \varphi_y^{-1}(z) = -\frac{1}{\eta}\ln(z^{-1/h} - 1), \tag{13}$$

where ($z^{-1/h}$-1)>0. With simple derivations, we obtain:

$$|\frac{\partial \varphi_y^{-1}(z)}{\partial z}| = \frac{1}{\eta h} \cdot \frac{1}{z(1 - z^{(1/h))}}. \tag{14}$$

In addition, $p(y)$ follows the standard normal distribution, which can be formulated as

$$p(y) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{y^2}{2}}. \tag{15}$$

By substituting Eq.13 into Eq.15, and then substituting Eqs.14-15 into Eq.12, we finally obtain

$$p(z) = \frac{1}{\sqrt{2\pi}h\eta} \cdot \frac{1}{z(1 - z^{1/h})} \cdot \mathrm{e}^{-\frac{(\ln(z^{(-1/h)} - 1))^2}{2\eta^2}}, \tag{16}$$

which completes the proof.

Appendix B
The Controller on Other Typical Distributions in Sec.3.4.3

Now we consider the effects of the distribution controller beyond normal distributions. Of note, $x \in \mathbf{X}$ denotes the input of the controller (output of the generator), $y \in \mathbf{Y}$ denotes the intermediate variable after instance normalization, and $z \in \mathbf{Z}$ denotes the output of the distribution controller. For convenience, we show the illustrative positions of these variables in Fig.12.

Recall that we have y=$\omega(x)$ via Eq.6 and z=$\varphi(y)$ via Eq.7. According to the probability density transformation [9], the transformed PDFs $p(y)$ and $p(z)$ can be obtained as

$$p(y) = p_x(\omega_x^{-1}(y)) \cdot |\frac{\partial \omega_x^{-1}(y)}{\partial y}| \tag{17}$$

and

$$p(z) = p_y(\varphi_y^{-1}(z)) \cdot |\frac{\partial \varphi_y^{-1}(z)}{\partial z}|, \tag{18}$$

where $\omega_x^{-1}(y)$ is the inverse function of y on x, and $\varphi_y^{-1}(z)$ is the inverse function of z on y. In particular, since $\omega(x)$ in Eq.6 denotes the instance normalization, we obtain

$$x = \omega_x^{-1}(y) = y\sqrt{\mathbb{V}[x]} + \mathbb{E}[x], \quad |\frac{\partial \omega_x^{-1}(y)}{\partial y}| = \sqrt{\mathbb{V}[x]}. \tag{19}$$

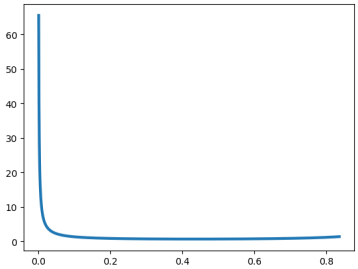


Fig. 13. The transformed PDF curve for inputs with uniform distributions.

B.1 On Uniform Distributions

Firstly, we consider the variable x with uniform distributions, whose PDF can be formulated as

$$p(x) = \begin{cases} \dfrac{1}{b-a}, & a < x < b \\ 0 \quad , \text{ others.} \end{cases} \tag{20}$$

For uniform distributions, we can calculate the expectation $\mathbb{E}[x]$ as $\frac{a+b}{2}$, and the variance $\mathbb{V}[x]$ as $\frac{(b-a)^2}{12}$. According to Eq.19, we obtain

$$x = \omega_x^{-1}(y) = \frac{b-a}{2\sqrt{3}} \times y + \frac{a+b}{2} \tag{21}$$

and

$$|\frac{\partial \omega_x^{-1}(y)}{\partial y}| = \frac{b-a}{2\sqrt{3}}. \tag{22}$$

By substituting Eqs.20-22 into Eq.17, $p(y)$ is obtained as

$$p(y) = \begin{cases} \dfrac{1}{2\sqrt{3}}, & a < \dfrac{b-a}{2\sqrt{3}} \times y + \dfrac{a+b}{2} < b \\ 0 \quad , \text{ others,} \end{cases} \tag{23}$$

which is equal to

$$p(y) = \begin{cases} \dfrac{1}{2\sqrt{3}}, & -\sqrt{3} < y < \sqrt{3} \\ 0 \quad , \text{ others.} \end{cases} \tag{24}$$

Furthermore, by substituting Eqs.13-14 and Eq.24 into Eq.18, we obtain $p(z)$ as

$$\begin{cases} \dfrac{1}{2\sqrt{3}\eta h z(1-z^{(1/h)})}, -\sqrt{3} < -\dfrac{1}{\eta}\ln(z^{-1/h}-1) < \sqrt{3} \\ 0 \quad , \text{others,} \end{cases} \tag{25}$$

which is equal to

$$\begin{cases} \dfrac{1}{2\sqrt{3}\eta h z(1-z^{(1/h)})}, (\mathrm{e}^{\sqrt{3}\eta}+1)^{-h} < z < (\mathrm{e}^{-\sqrt{3}\eta}+1)^{-h} \\ 0 \quad , \text{others.} \end{cases} \tag{26}$$

With η=1.5 and h=2.5 in Eq.26, we finally obtain the PDF curve of $p(z)$, as shown in Fig.13.

From this figure, we observe that $p(z)$ has a clear right tail over (0,1). Since we aim to obtain right-skewed distributions on the final output z, this observation demonstrates the effectiveness of our controller for the inputs with uniform distributions.

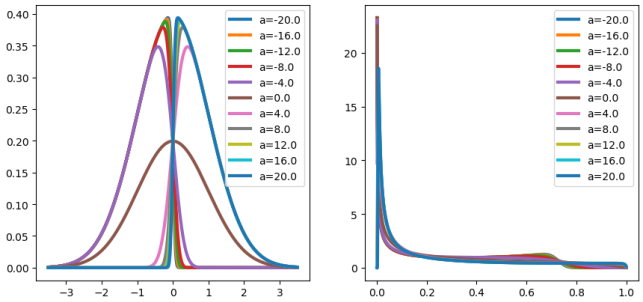


Fig. 14. The transformed PDF curves for inputs with skew normal distributions. The left shows the original distributions, and the right shows the transformed distributions.

B.2 On Skew Normal Distributions

Below we analyze the effects on skewed distributions. For simplicity, we focus on the skew normal distribution, which can be formulated as

$$p(x) = 2\phi(x)\Phi(ax), \tag{27}$$

where $\phi(x)=\frac{1}{\sqrt{2\pi}}\mathrm{e}^{(-\frac{x^2}{2})}$, $\Phi(x)=\int_{-\infty}^{x}\phi(t)\mathrm{d}t=\frac{1}{2}[1+\mathrm{erf}(\frac{x}{\sqrt{2}})]$ (erf denotes "error function"). Similarly, by substituting Eq.19 and Eq.27 into Eq.17, we can obtain $p(y)$:

$$2\phi(y\sqrt{\mathbb{V}[x]}+\mathbb{E}[x])\Phi(a(y\sqrt{\mathbb{V}[x]}+\mathbb{E}[x]))\times\sqrt{\mathbb{V}[x]}, \tag{28}$$

where we have $\mathbb{E}[x]=\frac{\sqrt{2}a}{\sqrt{(1+a)\pi}}$ and $\mathbb{V}[x]=(1-\frac{2a^2}{\pi(1+a^2)})$ for the skew normal distributions. By substituting Eqs.13-14 and Eq.28 into Eq.18, we obtain the transformed PDF $p(z)$:

$$\begin{aligned} &2\phi(\varphi_y^{-1}(z)\sqrt{\mathbb{V}[x]}+\mathbb{E}[x])\Phi(a(\varphi_y^{-1}(z)\sqrt{\mathbb{V}[x]}+\mathbb{E}[x])) \\ &\times\sqrt{\mathbb{V}[x]}\times|\frac{\partial\varphi_y^{-1}(z)}{\partial z}|, \end{aligned} \tag{29}$$

where $\varphi_y^{-1}(z)=-\frac{1}{\eta}\ln(z^{-1/h}-1)$ and $|\frac{\partial\varphi_y^{-1}(z)}{\partial z}|=\frac{1}{\eta h z(1-z^{(1/h)})}$. We substitute η=1.5 and h=2.5 into Eq.29 to obtain the final PDF. In particular, we change a from -20 to 20, and display their corresponding transformed PDF curves in Fig.14.

As we can see, for the different skewness of the original distributions (the right-skewed distribution with $a>0$ or the left-skewed distribution with $a<0$), their corresponding transformed distributions $p(z)$s have clear right tails over (0,1). As before, it demonstrates the effectiveness of the proposed controller for the inputs with skew normal distributions.

B.3 On Mixture of Normal Distributions

Now we consider the mixture of normal distributions. Suppose $\Psi(x,\sigma_i,\mu_i)=\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}$ and r_i denotes the weight of the i-th component with $\sum_i r_i$=1. The PDF for x with the mixture of normal distributions is formulated as

$$p(x) = \sum_i r_i \Psi(x,\sigma_i,\mu_i). \tag{30}$$

By substituting Eq.19 and Eq.30 into Eq.17, we obtain

$$p(y) = \sum_i r_i \Psi(y\sqrt{\mathbb{V}[x]}+\mathbb{E}[x],\sigma_i,\mu_i)\sqrt{\mathbb{V}[x]}. \tag{31}$$

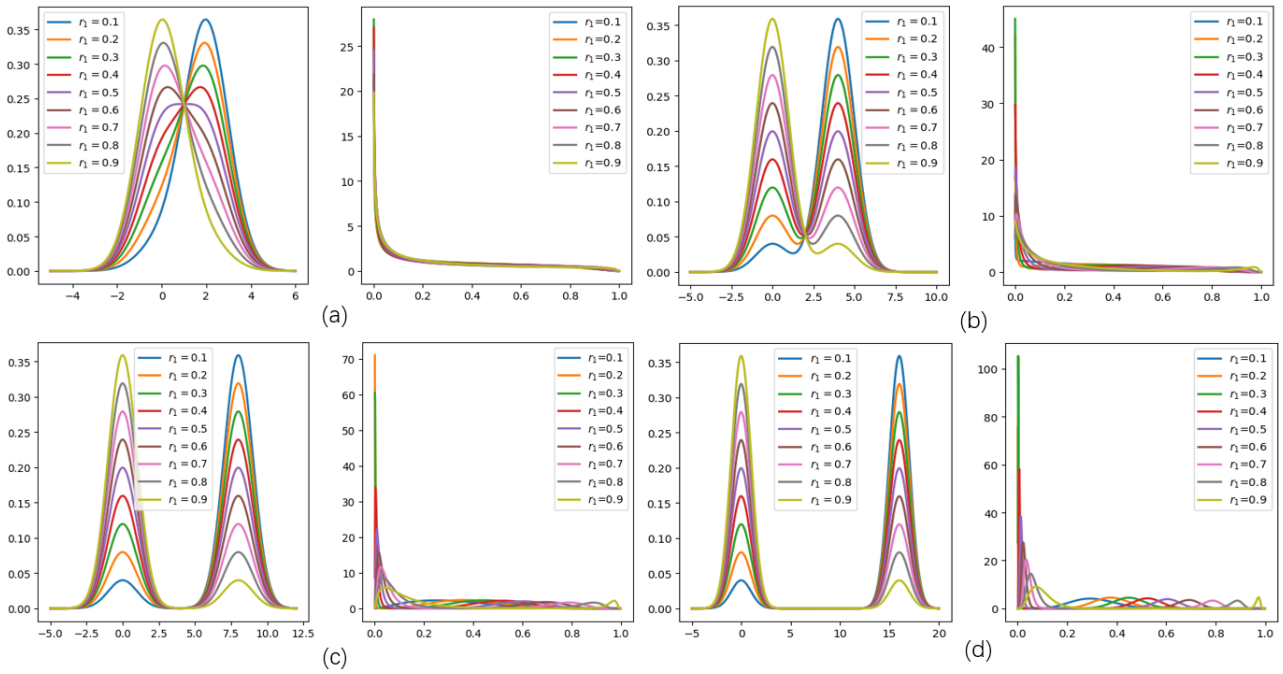


Fig. 15. The transformed PDF curves for inputs with the mixture of normal distributions. The left shows the original distributions, and the right shows the transformed distributions. Specifically, for these sub-figures, we have (a) $\mu_2 = 2$, (b) $\mu_2 = 4$, (c) $\mu_2 = 8$, (d) $\mu_2 = 16$.

In particular, we focus on the mixture of two normal distributions, where we have $\mathbb{E}(x)$=$r_1\mu_1$+$r_2\mu_2$ and $\mathbb{V}[x]$=$r_1\sigma_1^2$+$r_2\sigma_2^2$+$r_1r_2(\mu_1$-$\mu_2)^2$. Similarly, we substitute Eqs.13-14 and Eq.31 into Eq.18 and obtain $p(z)$ as

$$\sum_i r_i \Psi(\varphi_y^{-1}(z)\sqrt{\mathbb{V}[x]} + \mathbb{E}[x], \sigma_i, \mu_i)\sqrt{\mathbb{V}[x]}|\frac{\partial \varphi_y^{-1}(z)}{\partial z}|, \quad (32)$$

where $\varphi_y^{-1}(z)$=$-\frac{1}{\eta}\ln(z^{-1/h}$-1) and $|\frac{\partial \varphi_y^{-1}(z)}{\partial z}|$=$\frac{1}{\eta h z(1-z^{(1/h)})}$.

We set η=1.5 and h=2.5 in Eq.32 as before to build the final transformed PDF. For simplicity, we set σ_1=σ_2=1 and μ_1=0. We change r_1 within $\{0.1, 0.2, \ldots, 0.9\}$ and μ_2 within $\{2, 4, 8, 16\}$. The curves of the corresponding transformed PDFs are displayed in Fig.15. Of note, the mixtures of the above normal distributions with r_1=$\{0, 1\}$ or μ_2=0 equal to single normal distributions.

From this figure, we can obtain the following observations. Firstly, a large μ_2 brings a significant characteristic of bimodal distributions, namely two distinct peaks, which increases the difficulty of transformation. For example, with μ_2=16 and r_1=0.9, a part of values between (0.2,0.8) will be wasted due to the extremely low probabilities. However, if the two distributions are not too far away from each other (with respect to their variances), we observe that only a minority of the transformed scores are close to 1, and a majority of the scores are much lower and different. Thus, although this figure does not cover all kinds of mixtures of normal distributions, it still shows the effectiveness of our controller on the mixtures to some degree. Secondly, small r_1s result in the left tails for the original distributions. Benefiting from our controller, their transformed distributions are guided towards the right-skewed ones.

APPENDIX C
THE VARIANTS OF SELF-BASED TUNING

In DRE+ (Self-based), we take advantage of the ranking of scores and use the accumulated class probabilities to improve masks. However, its post hoc trick is independent of the model training. Below we propose potential variants of the self-based tuning with an end-to-end optimization, which could probably help the optimization of the proposed relevance estimator and make it more adaptable. We leave the experimental investigations as future work.

Note that during the self-based tuning, we first estimate the extra weight W_k=$Q_{\max}$-Q_k for each pixel, where $Q_{\max}$ denotes the maximal probability and Q_k is the current accumulated probability. Then we combine it with the original score M_k for the final relevance score in the saliency map, represented as M'_k=$M_k \times W_k$. More details can be found in Sec. 4.4.1. Although the variables W_ks are untrainable for a fixed DNN, they keep being updated during the optimization of M_k. Therefore, we can directly introduce $\mathbf{M}'$ as the mask in Eq.9 for model training. Compared with the original version where all W_ks are equal to 1, it weakens the obtained relevance scores, especially the pixels within the background. Since the number of high scores is limited for each mask, this variant could enforce the relevance estimator to take more effort to detect important supporting pixels while ignoring the role of background, which further improves the differentiation. The potential issue is that calculating all W_ks for each image can lead to huge time costs in model training. Therefore, the trade-off between the performance and the training efficiency built upon the linear interpolation of W_ks needs further investigations. Further improvement can be made for the generation of W_ks. That is, to avoid the considerable time costs of estimating W_ks for testing images, we can predict them by learning to fit W_ks

of training data. However, it inevitably introduces an extra hyper-parameter in the objective and requires more effort for sensitive analysis.

ACKNOWLEDGEMENT

We are grateful to the associate editor and the reviewers for their great efforts in improving the quality of this paper. This work is partially supported by the National Nature Science Foundation of China under grants 61725203, 62020106007 and 61772171, the Fundamental Research Funds for the Central Universities under grants PA2020GDKC0023 and PA2019GDZC0095, and the China Scholarship Council.

REFERENCES

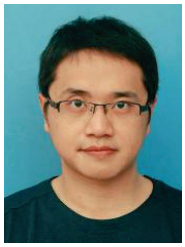
[1] L. Azzopardi, P. Thomas, and A. Moffat, "cwl_eval: An evaluation tool for information retrieval," in *Proceedings of SIGIR*, 2019, pp. 1321–1324.
[2] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proceedings of WACV*. IEEE, 2018, pp. 839–847.
[3] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," *SIAM review*, vol. 51, no. 4, pp. 661–703, 2009.
[4] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," in *Proceedings of NeurIPS*, 2017, pp. 6967–6976.
[5] D. P. Doane and L. E. Seward, "Measuring skewness: a forgotten statistic?" *Journal of statistics education*, vol. 19, no. 2, 2011.
[6] M. Du, N. Liu, Q. Song, and X. Hu, "Towards explanation of dnn-based prediction with guided feature inversion," in *Proceedings of SIGKDD*, 2018.
[7] R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," in *Proceedings of ICCV*, 2019, pp. 2950–2958.
[8] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of ICCV*, 2017, pp. 3429–3437.
[9] C. Forbes, M. Evans, N. Hastings, and B. Peacock, "Statistical distributions," *John Wiley & Sons*, 2011.
[10] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE TNNLS*, vol. 28, no. 10, pp. 2222–2232, 2016.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR*, 2016, pp. 770–778.
[12] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.
[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of ICML*, 2015, pp. 448–456.
[14] B. Kulis *et al.*, "Metric learning: A survey," *Foundations and trends in machine learning*, vol. 5, no. 4, pp. 287–364, 2012.
[15] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of NeurIPS*, 2017, pp. 4765–4774.
[16] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.
[17] M. T. Ribeiro, S. Singh, and C. Guestrin, "'why should i trust you?' explaining the predictions of any classifier," in *Proceedings of SIGKDD*, 2016, pp. 1135–1144.
[18] M. T. Ribeiro., S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of AAAI*, 2018.
[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of ICCV*, 2017, pp. 618–626.
[20] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *Proceedings of ICLR Workshop*, 2014.
[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[22] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *ICML Workshop*, 2017.
[23] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *Proceedings of ICLR Workshop*, 2014.
[24] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of ICML*. JMLR. org, 2017, pp. 3319–3328.
[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of CVPR*, 2015, pp. 1–9.
[26] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.
[27] G. Waissi and D. F. Rossin, "A sigmoid approximation of the standard normal integral," *Applied Mathematics and Computation*, vol. 77, no. 1, pp. 91–95, 1996.
[28] J. Wu, M. Poloczek, A. G. Wilson, and P. Frazier, "Bayesian optimization with gradients," in *Proceedings of NeurIPS*, 2017, pp. 5267–5278.
[29] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks."
[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *Proceedings of ICLR*, 2015.

Weijie Fu is currently working toward the PhD degree in the School of Computer and Information, Hefei University of Technology (HFUT). His current research interest focuses on data mining and interpretable machine learning.

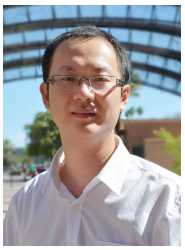
Meng Wang is a professor at the Hefei University of Technology, China. He received his B.E. degree and Ph.D. degree in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, in 2003 and 2008, respectively. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He has authored more than 200 book chapters, journal and conference papers in these areas. He is the recipient of the ACM SIGMM Rising Star Award 2014. He is an associate editor of IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), IEEE Transactions on Circuits and Systems for Video Technology (IEEE TCSVT), and IEEE Transactions on Neural Networks and Learning Systems (IEEE TNNLS).

Mengnan Du is currently a Ph.D. student at the Department of Computer Science and Engineering of Texas A&M University (TAMU). His research interests include interpretable machine learning and fairness in deep learning.

Ninghao Liu is a Ph.D. student in Computer Science at the Department of Computer Science and Engineering of Texas A&M University (TAMU). His research interests include explainable artificial intelligence, network embedding and anomaly detection.



Shijie Hao is an associate professor at School of Computer Science and Information Engineering, Hefei University of Technology (HFUT). He is also with Key Laboratory of Knowledge Engineering with Big Data (Hefei University of technology), Ministry of Education. He received his Ph.D. degree at HFUT in 2012. His research interests include image processing and pattern recognition.



Xia Hu is an associate professor and Lynn '84 and Bill Crane '83 Faculty Fellow at Texas A&M University (TAMU) in the Department of Computer Science and Engineering. He directs the Data Analytics at Texas A&M (DATA) Lab, and has published over 100 papers in several major academic venues, including KDD, WWW, SIGIR, IJCAI, AAAI, etc. His papers have received several awards, including WWW 2019 Best Paper Shortlist, INFORMS 2019 Best Poster Award, WSDM 2013 Best Paper Shortlist, IJCAI 2017 BOOM workshop Best Paper Award. He is the recipient of JP Morgan AI Faculty Award, Adobe Data Science Award, NSF CAREER Award, and ASU President Award for Innovation. He was the conference General Co-Chair for WSDM 2020.