Logic Rule Guided Attribution with Dynamic Ablation

Jianqiao An[1,2], **Yuandu Lai**[1,2,3,*], **Yahong Han**[1,2,†]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Tianjin Key Lab of Machine Learning, Tianjin University, Tianjin, China
[3]Peng Cheng Laboratory, Shenzhen, China
{anjianqiao, yuandulai, yahong}@tju.edu.cn

Abstract

With the increasing demands for understanding the internal behaviors of deep networks, Explainable AI (XAI) has been made remarkable progress in interpreting the model's decision. A family of attribution techniques has been proposed, highlighting whether the input pixels are responsible for the model's prediction. However, the existing attribution methods suffer from the lack of rule guidance and require further human interpretations. In this paper, we construct the 'if-then' logic rules that are sufficiently precise locally. Moreover, a novel rule-guided method, dynamic ablation (DA), is proposed to find a minimal bound sufficient in an input image to justify the network's prediction and aggregate iteratively to reach a complete attribution. Both qualitative and quantitative experiments are conducted to evaluate the proposed DA. We demonstrate the advantages of our method in providing clear and explicit explanations that are also easy for human experts to understand. Besides, through the attribution on a series of trained networks with different architectures, we show that more complex networks require less information to make a specific prediction.

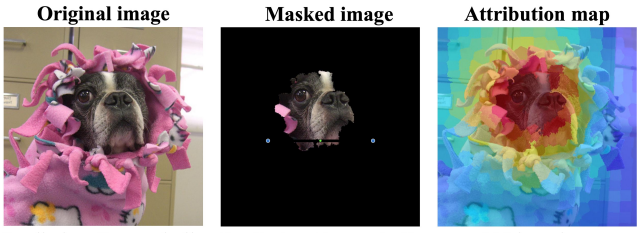


Figure 1: Guided by the constructed logic rules, our DA finds the minimal and sufficient part of the input to justify the classification. The clear part of the middle image (represented by P_{pos}) provides the explanation as 'If the part P_{pos} of the input x present, then x is classified as class Boston bull'. The area of P_{pos} is minimized by the DA algorithm iteratively, and the attribution map is generated at the same time (the rightmost image).

Introduction

Deep Neural models like Convolutional Neural Networks (CNNs) have achieved the state-of-the-art performance in different computer vision tasks. However, it is difficult to explain their predictions due to the lack of interpretability. A critical issue called attribution is to explain why classification CNNs predict what they predict (Selvaraju et al. 2017).

The attribution result is usually represented as a saliency map, i.e., a heatmap that highlights the input pixels that are the evidence for and against the classification outputs (Montavon, Samek, and Müller 2018). Suppose we have a model that predicts some kinds of lesions from an image of the organ (e.g., X-ray image or nuclear magnetic resonance image). The attribution maps identify the importance of each pixel to the prediction. Consequently, we not only get the diagnosis result but also know which part of the image that the model considers to be essential to the result. We can debug the model with the intervention of a professional doctor through the attribution maps. Attribution techniques could also assist the doctor to make a pathological diagnosis and improve diagnostic accuracy.

The existing methods of visual attribution can be broadly categorized into gradient-based (Simonyan, Vedaldi, and Zisserman 2014; Zeiler and Fergus 2014; Springenberg et al. 2014; Smilkov et al. 2017), perturbation-based (Petsiuk, Das, and Saenko 2018; Fong, Patrick, and Vedaldi 2019), and CAM-based methods (Selvaraju et al. 2017; Muhammad and Yeasin 2020; Wang et al. 2020; Fu et al. 2020). Gradient-based methods backpropagate the gradient of a target class to the input layer to highlight the image region that highly influences the prediction. In contrast, CAM-based methods highlight objects by resorting to the activation of feature maps. However, Adebayo et al. (Adebayo et al. 2018) found that a series of gradient-based and CAM-based attribution methods are insensitive to randomizing labels or model parameters. Thus, they cannot possibly explain mechanisms that depend on the relationship between instances and labels, such as an edge detector. As gradients are computed just as a direction of the increase on the loss function, pixels that cause high activation or with large gradients may not be sufficient enough to represent the 'necessary part' for the model to make predictions. Although these methods

*This work was done during Yuandu Lai's internship at Peng Cheng Laboratory.
†Corresponding author

are useful for the family of weakly supervised tasks, we argue that they are not explicit for interpreting deep models. Furthermore, some of the evaluation metrics (e.g., Pointing Game (Zhang et al. 2018)) of attribution methods are based on 'whether the attribution map is consistent with human understanding' (i.e., the pixels on the object is more important). To explain the model itself, we explore the necessary conditions (e.g., parts of the input image) that are sufficient to yield a specified prediction, rather than encouraging to get the attribution that people envisioned.

Zhang et al. (Zhang et al. 2020) recognize four major types of explanations, logic rules, hidden semantics, attribution, and explanations by examples listed in order of decreasing explanatory power. Logic rules provide the most explicit and clearest explanations (Zhang et al. 2020) while attribution provides the best visualization. In this work, we use the basic form of a logic rule *'if P, then Q'* to guide the attribution, where *P* is called the antecedent (e.g., a combination of several input features), and *Q* is called the consequent (e.g., prediction of a network). We construct logic rule to find what should be minimally and sufficiently present to justify its classification and analogously what should be minimally absent to confuse the classifier. In short, the explanation takes the form 'If the parts $p_1, ..., p_k$ of the input x are present (or absent), then x is classified as class y (or not y)'. Similar logic rules are proposed in (Dhurandhar et al. 2018).

To obtain *P* (i.e., $p_1, ..., p_k$) as well as the attribution map, we proposed a novel rule-guided algorithm called dynamic ablation (DA). We use several circles (called dynamic circle) to locate $p_1, ..., p_k$, and the total area of the circles is iteratively shrunk to ensure the consequent *Q* is satisfied (i.e., the network still makes the same prediction as it on the original input x). At each iteration, dynamic circles are encouraged to be smaller, and those that meet the rule are retained; otherwise, they are discarded. Such rule-guided explanations are lucid and easily understandable by humans. Furthermore, we attribute several trained networks with different architectures and find it interesting that more complex networks need less information to make a specific prediction.

Related Work

In this section, we mainly introduce prior works of logic rules and attribution.

Rule-Form Explanations

Logic rules are commonly acknowledged to be interpretable and have a long history of research. Most of the rule extraction methods provide global explanations as they only extract a single rule set or decision tree from the target model. Some of them make use of the network-specific information which are called decompositional approaches in previous literature (Craven and Shavlik 1994). Decompositional approaches generate rules by observing the connections in a network. One of the earliest methods is the KT algorithm (Fu 1991), which divides the input attributes into two groups, pos-atts (short for positive attributes) and neg-atts, according to the signs of their corresponding weights.

There are only a few methods producing local rule-form explanations for complex models. The explanation rules can be if-then, M-of-N, or some other forms such as the propositional rule, first-order rule, or fuzzy rule. Dhurandhar et al. (Dhurandhar et al. 2018) construct a local rule that provides contrastive explanations justifying the classification of an input. Wang et al. (Wang et al. 2018) came up with another local interpretability method that identifies critical data routing paths (CDRPs) of the network. However, these rule-form explanations suffer from a lack of intuitive visualization and may not be human-understandable.

Attribution Methods

Neural network attribution techniques can be broadly separated into three categories, gradient-based methods, CAM-based methods, and perturbation-based methods.

Gradient-Based Methods. In general, calculating the gradient of a model's output to the input features or the hidden neurons is the basis of this type of explanation method. Saliency maps proposed by Simonyan et al. (Simonyan, Vedaldi, and Zisserman 2014) use gradients to visualize relevant regions for a given class. As the generating saliency maps are usually noisy with vanilla gradients, subsequent methods (Springenberg et al. 2014; Montavon et al. 2017; Sundararajan, Taly, and Yan 2017; Nam, Lee et al. 2020; Zeiler and Fergus 2014; Smilkov et al. 2017) were developed to produce better visual heatmaps by modifying the gradient-based algorithms. However, gradients are computed just as a direction of the increase on the loss function, pixels with large gradients may not be sufficient enough to justify the model's predictions.

CAM-Based Methods. An extensive research effort (Selvaraju et al. 2017; Wang et al. 2020; Ramaswamy et al. 2020) has been put to blend high-level features extracted by CNNs in a unique explanation map based on the Class Activation Mapping (CAM) method (Zhou et al. 2016). Gradient-Weighted CAM (Grad-CAM) (Selvaraju et al. 2017) is a generalization of CAM that can target any layer and introduces the gradient information to CAM, which causes underestimation of sensitivity information due to gradient issues. Ablation-CAM (Ramaswamy et al. 2020) and Score-CAM (Wang et al. 2020) have been developed to overcome these drawbacks. Despite the strength of the CAM-based methods in capturing the features extracted in CNNs, the lack of localization information in the coarse high-level feature maps limits such methods' performance by producing blurry explanations (Sattarzadeh et al. 2020).

Perturbation-Based Methods. The perturbation-based approaches directly analyze the variations of the decision when distorting the input of the network. Few of these approaches, like RISE (Petsiuk, Das, and Saenko 2018), proposed random perturbation techniques to yield strong approximations of explanations. In Extremal Perturbation (EP) (Fong, Patrick, and Vedaldi 2019), an optimization problem is formulated to optimize a smooth perturbation mask maximizing the model's output confidence score. The problem with these methods is that they require exhaustive input modifications and suffer from a lack of rule guidance.

The Proposed Method

This section introduces the contrastive rules we construct and details the proposed rule-guided attribution method dynamic ablation (DA). First, we suggest minimal-positive rule and minimal-negative rule to provide contrastive explanations justifying the classification of an input by a black-box classifier. It is led to a ℓ_0 minimization problem and we perform input sampling with the proposed dynamic circles to solve such tasks. Finally, we optimize the time and space complexity of the method with superpixel technology.

Minimal-Positive Rule

A basic form of a logic rule is 'If P, then Q', where P is called the antecedent, and Q is called the consequent. For positive-rule:

$$\begin{aligned} P_{pos}&: \quad \text{parts of the input } x \text{ are } \textbf{present}, \\ Q_{pos}&: \quad x \text{ is classified as class } y, \end{aligned}$$

thus the explanation takes the form as 'if the parts $p_1, ..., p_k$ of the input x are present, then x is classified as a specific class y' (Pos-R for short). And our attribution method finds the minimal areas of $p_1, ..., p_k$, where k is the number of attributed parts.

For any input example $\boldsymbol{x} : \Omega \to \mathbb{R}^c$ and model Φ, we find an interpretable mask $\boldsymbol{m}$ assigning to each position $u \in \Omega$ a value $\boldsymbol{m}(u) = \{0, 1\}$ to conform to Pos-R, where Ω is an $h \times w$ discrete lattice. The aim is to find such $\boldsymbol{m}$ under the constraint that $\boldsymbol{m} \otimes \boldsymbol{x}$ meets Q_{pos} and has the fewest ones (i.e., sparsest solution). For each pixel $u \in \Omega$, $\boldsymbol{m}(u) = 1$ means that the pixel strongly contributes to the classification and $\boldsymbol{m}(u) = 0$ that it does not. The task leads to solving the following constrained ℓ_0 minimization problem:

$$\min_{\boldsymbol{m}: F(\boldsymbol{m} \otimes \boldsymbol{x}, Q_{pos}) = True} \|\boldsymbol{m}\|_0. \tag{1}$$

In problem (1), F is a Boolean function used to check whether $\boldsymbol{m} \otimes \boldsymbol{x}$ meets Q_{pos} (i.e., $\Phi(\boldsymbol{m} \otimes \boldsymbol{x}) = y$). It returns $True$ when it is satisfied, otherwise returns $False$. The pixels for which $\boldsymbol{m}(u) = 1$ are preserved, whereas the others are blacked out. It is worth mentioning that $F(\boldsymbol{x}, Q_{pos}) = True$ is satisfied as a premise in problem (1), because the class y in Q_{pos} is the prediction of the model Φ on original input x.

Minimal-Negative Rule

For the minimal-negative rule, we are interested in the parts of the input that most vulnerable to disturbance. It is defined as:

$$\begin{aligned} P_{neg}&: \quad \text{parts of the input } x \text{ are } \textbf{absent}, \\ Q_{neg}&: \quad x \text{ is } \textbf{not} \text{ classified as class } y, \end{aligned}$$

thus the explanation takes the form as 'if the parts $p_1, ..., p_k$ of the input x are absent, then x is not classified as a specific class y' (Neg-R for short). Analogously, our attribution method finds the minimal areas of $p_1, ..., p_k$ to confuse the classifier, and k is the number of attributed parts.

For any input example $\boldsymbol{x} : \Omega \to \mathbb{R}^c$ and model Φ, in contrast, we aim to find mask $\boldsymbol{m}$ under the constraint that $(\boldsymbol{J} - \boldsymbol{m}) \otimes \boldsymbol{x}$ meets Q_{neg} and has the fewest ones. In which,

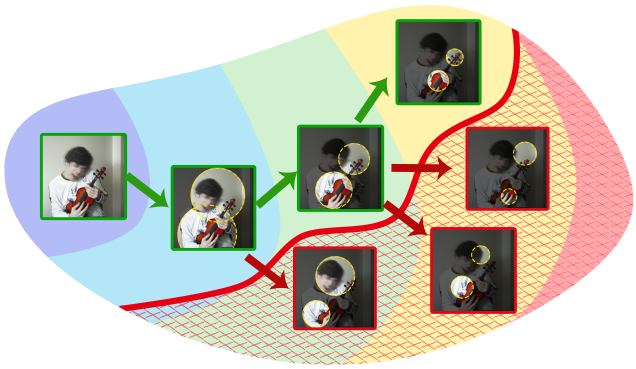


Figure 2: An illustration of DA guided by the rule Pos-R. As the model only sees the clear parts inside the circles, the images on the upper left side of the red curve are classified as a correct class 'ukulele', and the lower right images are not classified correctly. The background with different colors indicates the solution space of dynamic circles with different areas. We shrink the circles iteratively and retain the solutions that meet Pos-R (green arrows); otherwise, we make a new try (red arrows).

$\boldsymbol{J}$ represents a matrix of the same shape as $\boldsymbol{m}$ where each element equals 1. We formulate finding such mask $\boldsymbol{m}$ as the following constrained ℓ_0 minimization problem:

$$\min_{\boldsymbol{m}: F((\boldsymbol{J} - \boldsymbol{m}) \otimes \boldsymbol{x}, Q_{neg}) = True} \|\boldsymbol{m}\|_0. \tag{2}$$

In problem (2), the definition of the function F is the same as which in problem (1). Contrary to the problem (1), the pixels for which $\boldsymbol{m}(u) = 1$ are blacked out, whereas the others are preserved.

Dynamic Ablation

This subsection details the proposed rule-guided attribution algorithm. The optimization algorithm is based on random sampling from Ω, and the idea is inspired by the method of Decision-Based Attack (Brendel, Rauber, and Bethge 2018). They start with Gaussian noise and perform random sampling to reduce the noise close to the original image continuously to generate the adversarial sample. The combination of the source direction (pointing to the original image) and a random direction which is used as the sampling direction for each step, is theoretically proved effectively to minimize the ℓ_2 norm of noise. As the problem (1) and (2) are essential ℓ_0 minimization problems, we redefine the source direction as well as the random direction in our proposed algorithm.

We first initialize $\boldsymbol{m}$ to $\boldsymbol{J}$ as a start point and then iteratively reduce the elements of 1 in $\boldsymbol{m}$ according to the model decision. In each step, we change a small number of elements 1 to 0 in $\boldsymbol{m}$ and check whether the rule Pos-R (or Neg-R) is met. We retain the solution if it meets the rule; otherwise, we make a new try.

We propose to use k dynamic circles to search for $p_1, ..., p_k$ in the rule Pos-R or Neg-R. Each circle c can be defined as a triple (c_x, c_y, c_r), where (c_x, c_y) indicates the coordinate of its center and c_r indicates its radius. In this

Algorithm 1: Dynamic Ablation

Input: I, Φ
Output: $\boldsymbol{m}, \boldsymbol{cam}$

Initialize $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{r}$ randomly
$S_{bound} \leftarrow I.h \cdot I.w$
$\boldsymbol{m} \leftarrow \boldsymbol{J}, \boldsymbol{cam} \leftarrow \boldsymbol{0}$
for $itr = 0 \rightarrow itr\_number$ **do**
  $\boldsymbol{x'} \leftarrow \boldsymbol{x} + random(-I.h, I.h) \cdot \alpha_{step}$
  $\boldsymbol{y'} \leftarrow \boldsymbol{y} + random(-I.w, I.w) \cdot \alpha_{step}$
  $\boldsymbol{r'} \leftarrow \boldsymbol{r} + random(-\sqrt{\frac{S_{bound}}{\pi}}, \sqrt{\frac{S_{bound}}{\pi}}) \cdot \alpha_{step}$
  $\boldsymbol{r'} \leftarrow \frac{1}{\sqrt{\sum_{i=1}^{k} \boldsymbol{r_i'^2}}} \boldsymbol{r'} \cdot \frac{S_{bound}}{\pi}$
  $\boldsymbol{m'} \leftarrow GenerateMask(\boldsymbol{x'}, \boldsymbol{y'}, \boldsymbol{r'})$
  if $\Phi(\boldsymbol{m'} \otimes I) = \Phi(I)$ **then**
    $\boldsymbol{x} \leftarrow \boldsymbol{x'}$
    $\boldsymbol{y} \leftarrow \boldsymbol{y'}$
    $\boldsymbol{r} \leftarrow \boldsymbol{r'}$
    $\boldsymbol{m} \leftarrow \boldsymbol{m'}$
    $\boldsymbol{cam} \leftarrow \boldsymbol{cam} + \boldsymbol{m}$
    $S_{bound} \leftarrow S_{bound} \cdot f_c$
  end if
end for
$\boldsymbol{cam} \leftarrow Normalize(\boldsymbol{cam})$
return $\boldsymbol{m}, \boldsymbol{cam}$

way, we can obtain a unique mask $\boldsymbol{m}$ through a dynamic circle set $\mathbb{C}$:

$$\boldsymbol{m}_{x,y} = \begin{cases} 1, & Dis\left((x,y),(c_x,c_y)\right) \leq c_r, \forall c \in \mathbb{C} \\ 0, & Dis\left((x,y),(c_x,c_y)\right) > c_r, \forall c \in \mathbb{C}. \end{cases} \quad (3)$$

The function Dis in (3) is to calculate the two-dimensional Euclidean distance of two points. We convert the dimension of the optimization target from the original $h \times w$ to the circle-based $3 \times k$. The dimensionality reduction greatly improves the optimization efficiency and reduces the optimization difficulty.

Implementation Details. We use three k-dimensional vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{r}$ to represent the k circles in $\mathbb{C}$, and they walk randomly in the nearby range in each iteration under the control of the step rate. The randomly walking represents the random direction in the sampling. Furthermore, we iteratively reduce the sum area of the circles S_{bound} by multiplying a shrinkage factor f_c, and then normalize $\boldsymbol{r}$ so that $\pi \sum_{i=1}^{k} \boldsymbol{r}_i^2 = S_{bound}$ holds. The area reducing represents the source direction in the sampling. Mask $\boldsymbol{m}$ is generated corresponding to the circle set $\mathbb{C}$ according to (3). For the masks that meet the rule in the iterative process, we add them up and normalize them to get the attribution map. The attribution maps we generated are displayed in the form of heat-maps in Sec. . Refer to the Alg. 1 for more algorithm details.

In Alg. 1, $I.h$ and $I.w$ represent the height and width of the input image I. The function $GenerateMask$ is executed according (3), which is used to generate the 0/1-mask $\boldsymbol{m}$. And α_{step} represents the step rate mentioned above. Through the iteration, the sum area of the circles is ablated step by step, see Fig. 2 for a more intuitive description.

Complexity Optimization

It can be seen that for each iteration, the computational complexity of DA to implement sampling on an image with n pixels is $O(n)$. To make DA more efficient, we perform a superpixel segmentation algorithm to divide the original image into superpixels as preprocessing. Such algorithms are to divide an image into several fragments without intersecting, and each superpixel can be seen as a 'big pixel' in the downstream methods. Superpixels significantly reduce the number of original image pixels; thus, the computational complexity is also reduced to the same order as the numbers of superpixels.

We take the well-known SLIC (Achanta et al. 2012) algorithm to do superpixel segmentation in this work. According to the coordinates, we take the mean value of the pixels within each superpixel as the center coordinate of it. The superpixels whose center fall within any dynamic circles will be selected as a part of the attribution area. Consequently, we only traverse the superpixels in each iteration. The number of original pixels $h \times w$ can be much larger than the number of superpixels (hundreds of times). Therefore, the computation of associated optimization algorithms DA can be saved considerably. A runtime test was conducted to evaluate the computational complexity improvement by superpixels, the results are shown in Table. 2. It can be seen that the superpixel preprocessing can help to decrease the runtime by many orders of magnitude.

Experiment

Both qualitative and quantitative experiments are conducted respectively to evaluate the performance of different attribution methods. First, we qualitatively evaluate our method via visualization on ImageNet (Russakovsky et al. 2015) in Sec . Guided by the rule Pos-R and Neg-R, two ways of attribution are provided by our method. Second, we measure the attribution performance of different methods by inserting some parts of the input image with different sizes and calculate the classification accuracy. Compared with prior methods, the DA shows excellent performance advantages.

Furthermore, we design a novel experiment to compare the minimal necessary parts to justify different deep models. Experiments show that more complex networks need less information to make a specific prediction. Finally, an ablation study is carried to analyze the superpixel preprocessing effect on the running time.

Experimental Setup

In our experiments, we use pre-trained VGG16 (Simonyan and Zisserman 2015) and ResNet50 (He et al. 2016) networks from the PyTorch model zoo as base models. Publicly available object classification datasets, namely, ILSVRC2012 (Russakovsky et al. 2015) val and CIFAR100 (Krizhevsky, Nair, and Hinton 2009) are used as input images. We set the iteration number as 500, superpixel number as 500. The hyperparameters α_{step}, f_c, and k in our method are set to 0.2, 0.95, and 4, respectively. More analysis of such hyperparameters is shown in the ablation study section.

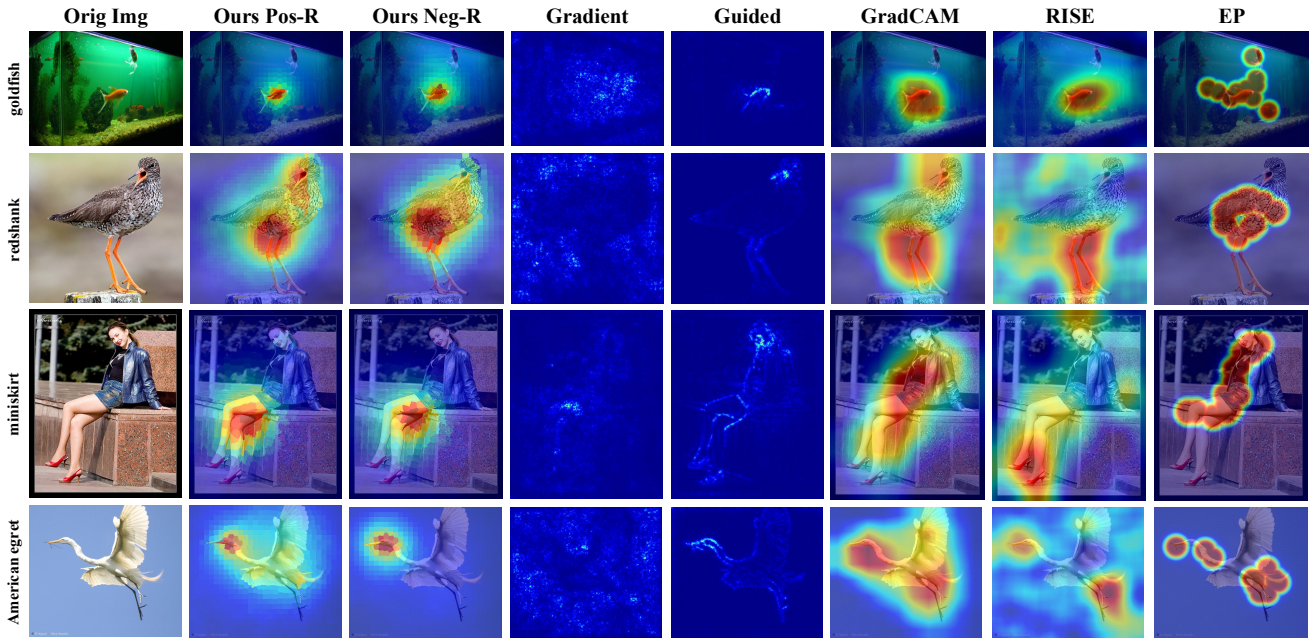


Figure 3: Qualitative comparison between various attribution methods applied to a pretrained ResNet-50 (He et al. 2016). From left to right: original image, ours guided by Pos-R, ours guided by Neg-R, Gradient (Simonyan, Vedaldi, and Zisserman 2014), Guided (Springenberg et al. 2014), GradCAM (Selvaraju et al. 2017), RISE (Petsiuk, Das, and Saenko 2018), EP (Fong, Patrick, and Vedaldi 2019).

Qualitative Assessment

This section provides the results of qualitative experiments of DA guided by two proposed rules Pos-R, Neg-R. Images are selected from ImageNet2012 (Russakovsky et al. 2015), and the ResNet50 (He et al. 2016) is used as the base model in the experiment. The visual attribution maps are displayed in the form of heat maps in Fig. 3, i.e., the red parts indicate the most important areas.

Guided by the rule Pos-R, we aim to find the minimally sufficient parts of the image to justify the model's prediction (column 2 in Fig. 3). As our attribution process is restricted by the rule Pos-R, the network always makes the same prediction as it on the original input, resulting our maps provide the most explicit and clearest explanations. For example, the image in the fourth row is classified as 'miniskirt'. Our map (column 2) shows that the base model pays special attention to leg features, not the miniskirt itself. It points out that the deep network based model's reasoning logic has a certain deviation from humans. However, other methods (Simonyan, Vedaldi, and Zisserman 2014; Springenberg et al. 2014; Selvaraju et al. 2017; Petsiuk, Das, and Saenko 2018; Fong, Patrick, and Vedaldi 2019) cannot reflect this deviation because of the lack of rules.

Contrarily, we aim to find the minimally sufficient parts of the image to disturb the model's prediction guided by the rule Neg-R (third column in Fig. 3). As shown in the results, the maps guided by the contrastive rules Pos-R and Neg-R roughly highlight the same parts. It shows that the class-discriminative parts (Pos-R) and the parts are easily to disturb (Neg-R) in such images are close. For the image on the second row, our method shows that both the mouth and legs features are necessary to justify the classification (Pos-R), while we just need to black out the legs area to make the model misclassify (Neg-R).

Minimum Attribution Ratio

In this section, we propose the concept of Minimum Attribution Ratio (MAR). For an input image $\boldsymbol{x}$ and a classification model Φ, we aim to find the sparsest $\boldsymbol{m}$ that meets criterion $\Phi(\boldsymbol{m} \otimes \boldsymbol{x}) = \Phi(\boldsymbol{x})$. The proportion of ones contained in the optimal $\boldsymbol{m}$ is defined as the Minimum Attribution Ratio (MAR). As our attribution process is always been restricted by the rule Pos-R, the final mask we found exactly represents the optimal $\boldsymbol{m}$. Through Fig. 3, it can be seen that for the same model Φ, the MAR corresponding to different input images are different. Comparably, experiments in this section show that for a specific input image, the MAR corresponding to the different models are also different. Furthermore, we find that more complex networks need less information to make a specific prediction.

We choose five well-known model architectures 'AlexNet (Krizhevsky, Sutskever, and Hinton 2017), ResNet18 (He et al. 2016), ResNet50 (He et al. 2016), VGG11 (Simonyan and Zisserman 2015), VGG16 (Simonyan and Zisserman 2015)' for classification tasks, and pretrain them on ImageNet2012. Then, we screen out 500 images that can be correctly classified by these five models from ImageNet2012 as the test set. We use the proposed

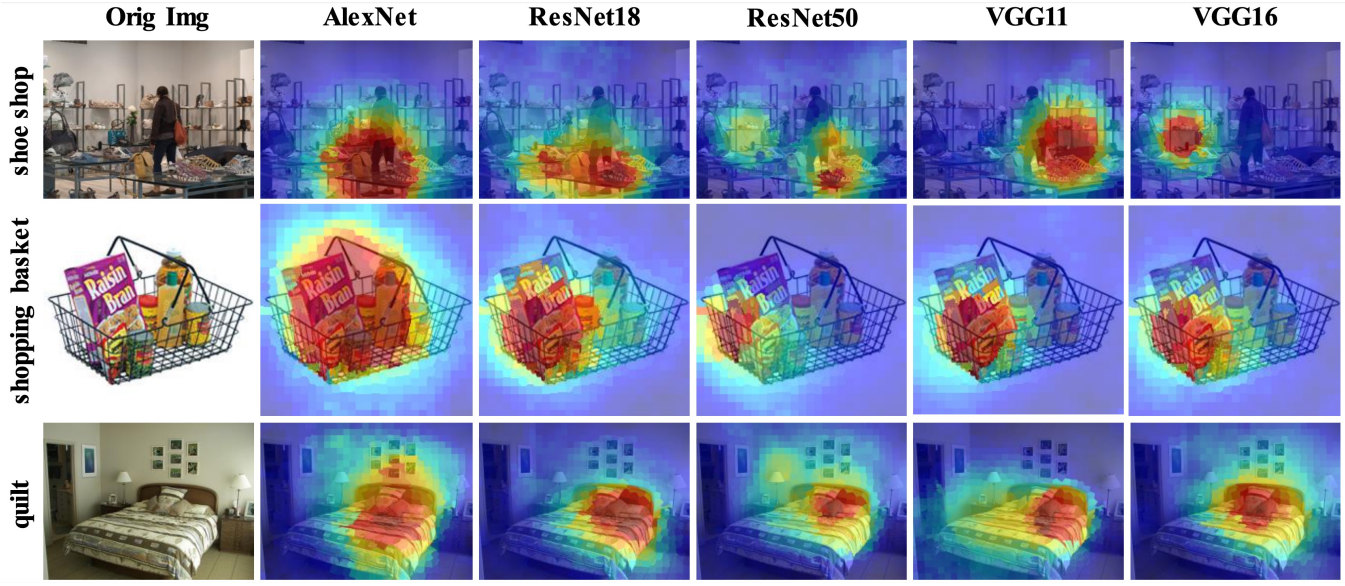
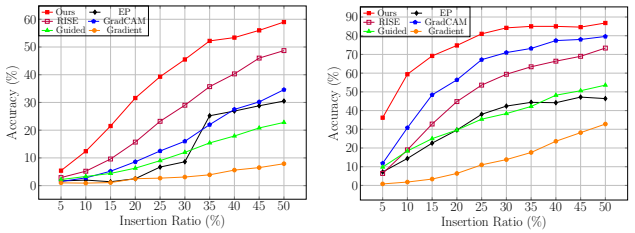


Figure 4: Visualization of attribution maps applied to different base models. From left to right: original image, AlexNet, ResNet18, ResNet50, VGG11, VGG16. The red part in each attribution map is the most critical area to justify the model's prediction.

Network	AlexNet	ResNet18	ResNet50	VGG11	VGG16
MAR	11.68%	5.31%	3.50%	4.56%	4.02%

Table 1: MAR corresponding to different model architectures. It shows that more complex networks need less information to make a specific prediction.



(a) CIFAR100 / VGG16 (b) ImageNet2012 / ResNet50

Figure 5: Quantitative results on attribution maps. The figure on the left side shows the results on CIFAR100/VGG16, and the right one shows the results on ImageNet2012/ResNet50. For all the methods, the corresponding classification accuracies are improving as the value of the insertion ratio increases. It is worth mentioning that when the insertion ratio $\geq 30\%$, the classification accuracy of our results on ImageNet2012 even exceeds the initial accuracy of 82.2%.

dynamic ablation algorithm to attribute the images of the generated test set with different base models.

With the model Φ, a test sample image $\boldsymbol{x}$, the dynamic ablation algorithm finds the mask $\boldsymbol{m}$ containing the least elements of 1. We calculate the ratio of the total number of 1 to the entire mask size $h \times w$ to get MAR. For all images in the test set, we average the MAR of all images as the final result of the model Φ. As shown in Table. 1, AlexNet as the simplest model architecture yields the largest MAR value. In the comparison of the two ResNet based architectures, ResNet18 needs more image information than ResNet50 to make decisions. Analogously, in the two VGG architectures, VGG11 has a larger MAR value than the deeper model VGG16. Overall, ResNet50 has the smallest MAR value 3.50% among these five models. In other words, on average, only 3.50% of the pixel information of each test image needs to be provided to ResNet50 for making a correct decision.

In addition, we generate visualization results for this part of the experiment. Fig. 4 shows the attribution maps of different models for three selected images. The more complex model architectures produce smaller red areas in the attribution maps. This is consistent with the results of the quantitative experiment in Table. 1.

Quantitative Evaluation

Motivated by (Fong and Vedaldi 2017), we use the idea of insertion game to measure the performance of several attribution methods quantitatively. We first use these methods to generate attribution maps for each image in the constructed test set. We sort the pixels of each test image according to the importance provided by the attribution map. We take the first p of input pixels as the insertion area (that is, keep them unchanged), while the other pixels are blacked out. This forms a new test set that has been modified. We then use the base model to retest the classification accuracy on the modified test set. Higher classification accuracy indicates the better performance of the corresponding attribution method.

	$sp\_num = 50$	$sp\_num = 100$	$sp\_num = 500$	$sp\_num = 1000$	$Original$
$k = 1$	0.162 / 20.17	0.165 / 20.24	0.108 / 23.55	0.133 / 25.06	0.117 / 1301.21
$k = 2$	0.162 / 20.67	0.125 / 20.83	0.100 / 23.56	0.124 / 26.96	0.106 / 1309.48
$k = 4$	0.158 / 20.74	0.111 / 21.10	0.094 / 23.80	0.124 / 28.74	0.125 / 1586.11
$k = 8$	0.162 / 21.40	0.176 / 21.43	0.130 / 24.63	0.127 / 29.08	0.137 / 1614.29
$k = 16$	0.172 / 23.32	0.266 / 23.89	0.170 / 29.16	0.153 / 35.69	0.183 / 2723.76

Table 2: Results of the MAR/runtime on DA with different values of k and sp_num. The runtimes are shown in seconds. The rightmost column shows the results of DA without superpixel preprocessing. It shows that when $sp\_num = 500$, superpixel preprocessing can not only improve the performance but also greatly decrease the running time.

We test attribution methods on two well-known datasets: CIFAR100 (Krizhevsky, Nair, and Hinton 2009) and ImageNet2012 (Russakovsky et al. 2015). We randomly select 1000 images from CIFAR100 and 500 images from ImageNet2012 as test samples. We use the pretrained VGG16 model as the base model for CIFAR100 and ResNet50 for ImageNet2012. On the original test set without any changes, the classification accuracy is CIFAR100/VGG16 - 68.9% and ImageNet2012/ResNet50 - 82.2%.

We test the attribution performance of different methods by setting different p values. The experimental results are shown in Fig. 5. The abscissa represents the proportion of different insertion areas (i.e., p), and the ordinate represents the classification accuracy on the modified test set. As our attribution process is always guided by the logic rules (i.e., the base model makes the correct prediction), the performance of our method is consequently higher than other methods, especially when p is small. The GradCAM (Selvaraju et al. 2017) shows good performance in ImageNet2012, but it is worse in CIFAR100. As our DA is guided by the logic rules and only the samples which yield correct prediction are retained, resulting the DA has superior performance on both datasets.

Complexity Evaluation. In addition to performance evaluations, a runtime test is carried out to compare the complexity of the methods, timing how long it took for each method to generate an explanation map. We use a GeForce GTX TITAN X GPU with 12GB of memory and the ResNet-50 as the base model in this experiment. We randomly select 500 images from ImageNet2012, and the reported runtimes were averaged over 500 trials. The gradient-based or CAM-based methods Grad-CAM (Selvaraju et al. 2017), Gradient (Simonyan, Vedaldi, and Zisserman 2014), and Guided (Springenberg et al. 2014) are the fastest methods which achieve the runtime 25.4, 38.1, and 107.2 milliseconds, respectively. On the other hand, RISE (Petsiuk, Das, and Saenko 2018) and EP (Fong, Patrick, and Vedaldi 2019) recorded pretty longer runtimes, 35.9 and 85.7 seconds, since they both perform iterations many times. In comparison with RISE (Petsiuk, Das, and Saenko 2018) and EP (Fong, Patrick, and Vedaldi 2019), DA runs in 23.8 seconds. It is worth mentioning that the superpixel processing we proposed reduces the search space and greatly improves the time efficiency, which is shown in Table. 2. The original methods in the rightmost column without superpixel optimization requires 10^3 second-level arithmetic processing, and through superpixel optimization, the time efficiency is improved by nearly a hundred times.

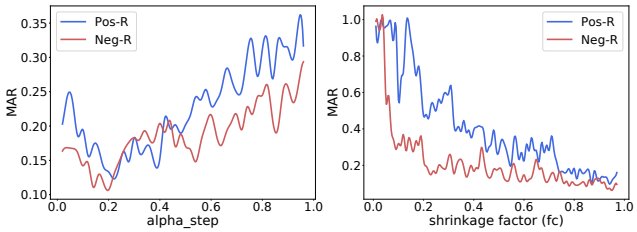


Figure 6: The figures show MAR curves with different values of α_{step} and f_c. The blue curve is guided by the rule Pos-R and the red one is guided by Neg-R. It can be seen that MAR takes the minimum near $\alpha_{step} = 0.2$ and $f_c = 0.95$.

Ablation Study. To analyze the effect of hyperparameters on DA's performance, an ablation study is carried. We use pretrained ResNet50 as the base model and randomly select 200 images from ImageNet2012 as input samples. By changing the α_{step} and f_c from 0 to 1 with 0.01 increment in each step, the mean MAR of the test images change accordingly. See Fig. 6 for more details. While the number of dynamic circles and superpixels k and sp_num also affect DA's performance, we show the differences of MAR and runtimes (shown in seconds) by setting different values of k, sp_num in Table. 2.

Conclusion

In this work, we propose the attribution method called dynamic ablation (DA) and construct two types of logic rules Pos-R and Neg-R, to guide the proposed method. We aim to find a minimal bound sufficient in an input image to justify the network's prediction and aggregate iteratively to reach a complete attribution. Different from the existing methods, the rule guided explanation does not need further human interpretations. Our method outperforms the previous attribution methods in providing explicit and clear explanations. Furthermore, we proposed the evaluation metric MAR and find that more complex networks need less information to make a specific prediction. Future work involves exploring the application of the proposed method in interpreting the meaning of hidden neurons or layers.

Acknowledgments

This work is supported by the NSFC (under Grant 61876130, 61932009)

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11): 2274–2282.

Adebayo, J.; Gilmer, J.; Muelly, M. C.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems*, volume 31, 9505–9515.

Brendel, W.; Rauber, J.; and Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *International Conference on Learning Representations*.

Craven, M. W.; and Shavlik, J. W. 1994. Using sampling and queries to extract rules from trained neural networks. In *Machine learning proceedings 1994*, 37–45. Elsevier.

Dhurandhar, A.; Chen, P.-Y.; Luss, R.; Tu, C.-C.; Ting, P.; Shanmugam, K.; and Das, P. 2018. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *arXiv preprint arXiv:1802.07623*.

Fong, R.; Patrick, M.; and Vedaldi, A. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2950–2958.

Fong, R. C.; and Vedaldi, A. 2017. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 3449–3457.

Fu, L. 1991. Rule Learning by Searching on Adapted Nets. In *AAAI*, volume 91, 590–595.

Fu, R.; Hu, Q.; Dong, X.; Guo, Y.; Gao, Y.; and Li, B. 2020. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. *arXiv preprint arXiv:2008.02312*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2009. CIFAR-100 (Canadian Institute for Advanced Research).

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of The ACM*, 60(6): 84–90.

Montavon, G.; Lapuschkin, S.; Binder, A.; Samek, W.; and Müller, K.-R. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65: 211–222.

Montavon, G.; Samek, W.; and Müller, K.-R. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73: 1–15.

Muhammad, M. B.; and Yeasin, M. 2020. Eigen-CAM: Class Activation Map using Principal Components. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–7. IEEE.

Nam, W.-J.; Lee, S.-W.; et al. 2020. Interpreting Deep Neural Networks with Relative Sectional Propagation by Analyzing Comparative Gradients and Hostile Activations. *arXiv preprint arXiv:2012.03434*.

Petsiuk, V.; Das, A.; and Saenko, K. 2018. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *BMVC*, 151.

Ramaswamy, H. G.; et al. 2020. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 983–991.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Sattarzadeh, S.; Sudhakar, M.; Lem, A.; Mehryar, S.; Plataniotis, K.; Jang, J.; Kim, H.; Jeong, Y.; Lee, S.; and Bae, K. 2020. Explaining Convolutional Neural Networks through Attribution-Based Input Sampling and Block-Wise Feature Aggregation. *arXiv preprint arXiv:2010.00672*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR 2015 : International Conference on Learning Representations 2015*.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. A. 2014. Striving for Simplicity: The All Convolutional Net. In *ICLR (workshop track)*.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*.

Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; and Hu, X. 2020. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 24–25.

Wang, Y.; Su, H.; Zhang, B.; and Hu, X. 2018. Interpret neural networks by identifying critical data routing paths. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8906–8914.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and Understanding Convolutional Networks. In *13th European Conference on Computer Vision, ECCV 2014*, 818–833.

Zhang, J.; Bargal, S. A.; Lin, Z.; Brandt, J.; Shen, X.; and Sclaroff, S. 2018. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10): 1084–1102.

Zhang, Y.; Tiňo, P.; Leonardis, A.; and Tang, K. 2020. A Survey on Neural Network Interpretability. *arXiv preprint arXiv:2012.14261*.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.