Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs

Ruigang Fu[1]
furuigang08@nudt.edu.cn
Qingyong Hu[2]
qingyong.hu@cs.ox.ac.uk
Xiaohu Dong[1]
dongxiaohu16@nudt.edu.cn
Yulan Guo[1]
yulan.guo@nudt.edu.cn
Yinghui Gao[1]
yhgao@nudt.edu.cn
Biao Li[1]
libiao_cn@163.com

[1] College of Electronic Science and Technology
National University of Defense Technology
Changsha, China

[2] Department of Computer Science
University of Oxford
Oxfordshire, UK

Abstract

To have a better understanding and usage of Convolution Neural Networks (CNNs), the visualization and interpretation of CNNs has attracted increasing attention in recent years. In particular, several Class Activation Mapping (CAM) methods have been proposed to discover the connection between CNN's decision and image regions. However, in spite of the reasonable visualization, most of these methods lack clear and sufficient theoretical support. In this paper, we introduce two axioms – *Sensitivity* and *Conservation* – to the visualization paradigm of the CAM methods. Meanwhile, a dedicated Axiom-based Grad-CAM (XGrad-CAM) is proposed to satisfy these axioms as much as possible. Experiments demonstrate that XGrad-CAM is an enhanced version of Grad-CAM in terms of sensitivity and conservation. It is able to achieve better visualization performance than Grad-CAM, while also be class-discriminative and easy-to-implement compared with Grad-CAM++ and Ablation-CAM. Code is available at https://github.com/Fu0511/XGrad-CAM.

1 Introduction

Due to the strong capability of feature learning, CNN-based approaches have achieved the state-of-the-art performance in numerous vision tasks such as image classification [9, 12, 25], object detection [10, 21] and semantic segmentation [14]. However, the interpretability of CNNs is often criticized by the community, as these networks usually look like complicated black boxes with massive unexplained parameters [5, 13, 29]. Therefore, it is highly desirable and necessary to find a way to understand and explain what exactly CNNs learned,

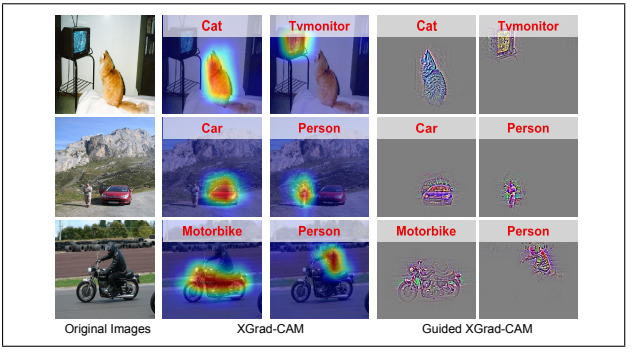


Figure 1: The visualization of our XGrad-CAM and Guided XGrad-CAM. It is clear that both of these two approaches are class-discriminative and able to highlight the object of interest. In addition, Guided XGrad-CAM provides more details than XGrad-CAM.

especially for applications where interpretability is essential (e.g., medical diagnosis and autonomous driving).

An important issue in CNN learning is to explain why classification CNNs predict what they predict [24]. Since both semantic and spatial information can be preserved in feature maps of deep layers, Gradient-weighted Class Activation Mapping (Grad-CAM) [24] was proposed to highlight important regions of an input image for CNN's prediction using deep feature maps. Specifically, Grad-CAM is defined as a linear combination of feature maps, where the weight of each feature map is determined by the average of its gradients. This definition is inspired by CAM [33] and further improved by other works, such as Grad-CAM++ [3] and Ablation-CAM [2]. However, most of these CAM methods lack clear theoretical support, e.g., why does Grad-CAM [24] use the average of gradients as the weight of each feature map?

In this paper, we propose a novel CAM method named XGrad-CAM (Axiom-based Grad-CAM) motivated by several formalized axioms. To achieve better visualization and explanation of CNN's decision, axioms are self-evident properties that visualization methods ought to satisfy [17, 23, 30]. Meeting these axioms makes a visualization method more reliable and theoretical. Therefore, two axiomatic properties are introduced in the derivation of XGrad-CAM: *Sensitivity* [30] and *Conservation* [17]. In particular, the proposed XGrad-CAM is still a linear combination of feature maps, but able to meet the constraints of those two axioms. The weight of each feature map in XGrad-CAM is defined as a weighted average of its gradients by solving an optimization problem. As shown in Fig. 1, our XGrad-CAM is a class-discriminative visualization method and able to highlight the regions belonging to the objects of interest. Further, by combining XGrad-CAM with the Guided Backprop [28], we propose Guided XGrad-CAM which provides more details of the objects than XGrad-CAM.

To summarize, the main contributions of this work are as follows:

- A dedicated XGrad-CAM with clear mathematical explanations is proposed to achieve better visualization of CNN's decision. It is able to be applied to arbitrary classification CNNs to highlight the objects of interest.

- By introducing two axioms as well as the corresponding axiom analysis, we can have a deeper understanding of why CAM methods work in visualizing the CNN's decision.

Our XGrad-CAM can be seen as an enhanced version of Grad-CAM in both sensitivity and conservation.

- Extensive experiments have been conducted to give a comprehensive comparison between the proposed XGrad-CAM and several recent CAM methods (i.e., Grad-CAM [24], Grad-CAM++ [3] and Ablation-CAM [4]). Taking both the class discriminability, efficiency and localization capability into consideration, our XGrad-CAM achieves better visualization performance.

2 Related Work

A number of methods have started to visualize the internal representations learned by CNNs [8, 17, 20] recently. These methods can be broadly categorized as: 1) visualization of filters and layer activations [2, 17], 2) visualization of hidden neurons [2, 15, 26, 32], 3) visualization of CNN's decision [24, 26, 32, 33]. In this section, we mainly introduce the visualization of CNNâĂŹs decision and some related axioms.

2.1 Visualization of CNN's Decision

These methods are developed to highlight the regions of an image, which are responsible for CNN's decision. They can be further categorized as: perturbation-based, propagation-based and activation-based methods.

(1) Perturbation-based methods. Zeiler et al. [32] occluded patches of an image using grey squares, and recorded the change of the class score. A heatmap can then be generated to show evidence for and against the classification. This method is further extended [34, 35] using different types of perturbations such as removing, masking or altering. While perturbation-based methods are straightforward, they are inefficient.

(2) Propagation-based methods. Propagation-based methods are rather fast, e.g., saliency maps proposed by Simonyan et al. [26] only require one forward propagation and one backward propagation through the network. Specifically, saliency maps use gradients to visualize relevant regions for a given class. However, with vanilla gradients, the generating saliency maps are usually noisy. Subsequent methods were developed to produce better visual heatmaps by modifying the back-propagation algorithm (e.g., Guided Backprop [28], Layerwise Relevance Propagation [1], DeepTaylor [16], Integrated Gradient[30], etc.) or averaging the gradients for an input with noise added to it [27].

(3) Activation-based methods. In contrast to propagation-based methods, activation-based methods highlight objects by resorting to the activation of feature maps. As an important branch of activation-based methods, CAM methods [3, 4, 18, 24] visualize CNN's decision using feature maps of deep layers. Zhou et al. [33] proposed the original CAM method which visualizes a CNN by linearly combining feature maps at the penultimate layer. The weight of each feature map is determined by the last layer's fully-connected weights corresponding to a target class. However, CAM is restricted to GAP-CNNs. That is, the penultimate layer is constrainted to be a global average pooling (GAP) layer. Selvaraju et al. [24] then proposed Grad-CAM to visualize an arbitrary CNN for classification by weighting the feature maps using gradients. Grad-CAM is inspired by CAM but hasn't explained its mechanism clearly (i.e., why using the average of gradients to weight each feature map). Aditya et al. [3] proposed Grad-CAM++ by introducing higher-order derivatives in Grad-CAM.

They assumed that the class score is a linear function of feature maps and got a closed-form solution of the weights for each feature map. Omeiza et al. [18] further proposed Smooth Grad-CAM++. This method follows the framework of Grad-CAM++ but uses SmoothGrad [27] to calculate gradients. More recently, Desai et al. [4] proposed Ablation-CAM to remove the dependence on gradients but this method is quite time-consuming since it has to run forward propagation for hundreds of times per image.

2.2 Axioms

For a visualization method of CNN's decision, axioms are properties that are considered to be necessary for the method. Existing axioms include continuity [17], implementation invariance [30], sensitivity [30] and conservation [17].

Given a model m, suppose that d features constitute an input $\mathbf{x}$, $f(\mathbf{x};m)$ represents a function of the model m w.r.t the input $\mathbf{x}$. The resulting explanation is denoted by $\mathbf{R}(\mathbf{x};m) \in \mathbb{R}^d$, where $R_i(\mathbf{x};m)$ represents the importance of the i-th feature for the model output.

Continuity is a property that if, for two nearly identical inputs, the model outputs are nearly identical, then the corresponding explanations should also be nearly identical, i.e., $\mathbf{R}(\mathbf{x};m) \approx \mathbf{R}(\mathbf{x}+\varepsilon;m)$ with ε a small perturbation.

Implementation invariance. Two models m_1 and m_2 are functionally equivalent if they produce the same output for any identical input. Implementation invariance requires to produce identical explanations for functionally equivalent models provided with identical input, i.e., $\mathbf{R}(\mathbf{x};m_1) = \mathbf{R}(\mathbf{x};m_2)$.

Sensitivity is a property that each response of the explanation should be equal to the output change caused by removing the corresponding feature of the input, i.e., $R_i(\mathbf{x};m) = f(\mathbf{x};m) - f(\mathbf{x}\backslash x_i;m)$ where the notation $\mathbf{x}\backslash x_i$ indicates a modified input where the i-th feature in the original input has been replaced by a baseline value (usually zero).

Conservation is a property that the sum of the explanation responses should match in magnitude of the model output, i.e., $f(\mathbf{x};m) = \sum_{i=1}^{d}(R_i(\mathbf{x};m))$.

In this paper, we mainly study the CAM methods using the axioms of sensitivity and conservation to visualize CNN's decision. Generally, gradient-based CAM methods violate the axiom of continuity because of the problem of shattered gradients [23]. Besides, they also break the axiom of implementation invariance since they are layer sensitive [4].

3 Approach

3.1 Notation and Motivation

Given an L-layer CNN and an input image $\mathbf{I}$, let l represent the index of the target layer for visualization, $\mathbf{F}^l$ denote the response of the target layer, $S_c(\mathbf{F}^l)$ represent the class score (the input to the softmax layer) of a class of interest c. Suppose that the l-th layer contains K feature maps, where the response of the k-th feature map is denoted as $\mathbf{F}^{lk}$. $F^{lk}(x,y)$ represents the response at position (x,y) in $\mathbf{F}^{lk}$.

To visualize the class c in the input image, a general form of the existing CAM methods [3, 4, 24] can be written as a linear combination of the feature maps in the target layer:

$$M_c(x,y) = \sum_{k=1}^{K}\left(w_c^k F^{lk}(x,y)\right), \tag{1}$$

where w_c^k is the weight of the corresponding feature map $\mathbf{F}^{lk}$, different definitions lead to different CAM methods. Then, to further identify the image regions responsible for the particular class c, two postprocessing are needed: the resulting map M_c needs to be ReLU rectified to filter negative units [24] and upsampled to the same size of the input image.

For the CAM methods, the key problem is how to precisely determine the importance of each feature map to the prediction of the class of interest. In this paper, we argue that it would be better if these CAM methods can satisfy two basic axioms, i.e., sensitivity and conservation.

Sensitivity: A general CAM method of Eq. (1) satisfies the axiom of sensitivity if it holds the following property for all the feature maps, that is:

$$S_c(\mathbf{F}^l) - S_c(\mathbf{F}^l \backslash \mathbf{F}^{lk}) = \sum_{x,y} \left(w_c^k F^{lk}(x,y) \right), \tag{2}$$

where $S_c(\mathbf{F}^l \backslash \mathbf{F}^{lk})$ is the score of class c when the k-th feature map in the target layer has been replaced by zero. This means that the importance of each feature map should be equivalent to the score change caused by its removing.

Conservation: To meet the axiom of conservation, a general CAM method of Eq. (1) should hold:

$$S_c(\mathbf{F}^l) = \sum_{x,y} \left(\sum_{k=1}^{K} \left(w_c^k F^{lk}(x,y) \right) \right). \tag{3}$$

This means that the responses of the CAM map should be a redistribution of the class score.

Intuitively, if a large drop of class score appears when we removed a specific feature map, this feature map would be expected as high importance. Sensitivity is exactly an axiom based on this intuition. Besides, conservation is introduced here to ensure that the class score can be mainly dominated by the feature maps rather than other unexplained factors. Therefore, introducing sensitivity and conservation is likely to make the CAM methods achieve more reasonable visualization.

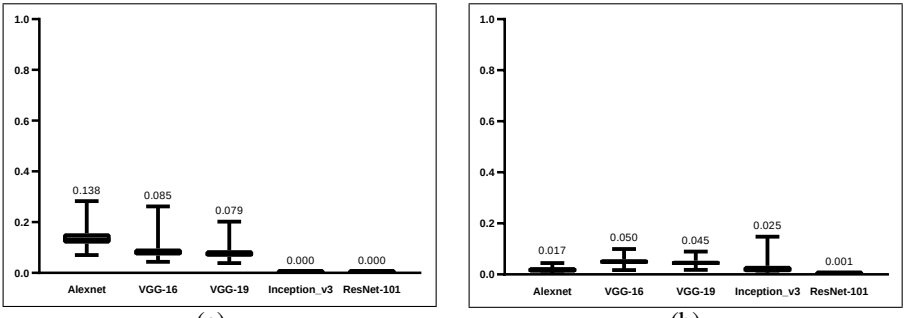
To meet the above two axioms as much as possible, we formulate this as a minimization problem of $\phi(w_c^k)$ as below:

$$\phi(w_c^k) = \underbrace{\sum_{k=1}^{K} \left| S_c(\mathbf{F}^l) - S_c(\mathbf{F}^l \backslash \mathbf{F}^{lk}) - \sum_{x,y} \left(w_c^k F^{lk}(x,y) \right) \right|}_{\text{Sensitivity}} + \underbrace{\left| S_c(\mathbf{F}^l) - \sum_{x,y} \left(\sum_{k=1}^{K} \left(w_c^k F^{lk}(x,y) \right) \right) \right|}_{\text{Conservation}}. \tag{4}$$

3.2 Our Method

Given an arbitrary target layer in a ReLU-CNN which only has ReLU activation as its non-linearities, we can prove that for any class of interest, the class score is equivalent to the sum of the element-wise products between feature maps and gradient maps of the target layer, followed with a bias:

$$S_c(\mathbf{F}^l) = \sum_{k=1}^{K} \sum_{x,y} \left(\frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk}(x,y)} F^{lk}(x,y) \right) + \varepsilon(\mathbf{F}^l), \tag{5}$$



(a) (b)

Figure 2: (a) Normalized $\zeta(\mathbf{F}^l;k)$ is small in the last spatial layers of different CNN models, including AlexNet [12], VGG-16 [25], VGG-19 [25], Inception_V3 [31] and ResNet-101 [9]; (b) Normalized $\varepsilon(\mathbf{F}^l)$ is also small in the last spatial layers of different CNN models. The mean values are provided above the box-plots.

where $\varepsilon(\mathbf{F}^l) = \sum_{t=l+1}^{L}\sum_j \frac{\partial S_c(\mathbf{F}^l)}{\partial u_j^t} b_j^t$, u_j^t denotes a unit in the layer t (t>l) and b_j^t is the bias term corresponding to the unit u_j^t. The detailed proof can be found in Appendix.

By further substituting the $S_c(\mathbf{F}^l)$ in Eq. (4) with the value in Eq. (5), we can have:

$$\begin{aligned}\phi(w_c^k) = \sum_{k=1}^{K}\left|\sum_{x,y}\left(\frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk}(x,y)}F^{lk}(x,y) - w_c^k F^{lk}(x,y)\right) + \zeta(\mathbf{F}^l;k)\right| + \\ \left|\sum_{k=1}^{K}\sum_{x,y}\left(\frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk}(x,y)}F^{lk}(x,y) - w_c^k F^{lk}(x,y)\right) + \varepsilon(\mathbf{F}^l)\right|.\end{aligned} \tag{6}$$

where $\zeta(\mathbf{F}^l;k) = \sum_{k'=1,k'\neq k}^{K}\sum_{x,y}\left(\frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk'}(x,y)}F^{lk'}(x,y) - \frac{\partial S_c(\mathbf{F}^l\backslash\mathbf{F}^{lk})}{\partial F^{lk'}(x,y)}F^{lk'}(x,y)\right) + \varepsilon(\mathbf{F}^l) - \varepsilon(\mathbf{F}^l\backslash\mathbf{F}^{lk})$. For the terms $\zeta(\mathbf{F}^l;k)$ and $\varepsilon(\mathbf{F}^l)$, they are difficult to optimize by the variable w_c^k since there are no direct relationship between these terms and the k-th feature map of the target layer. As a workaround, we calculated their normalized versions (i.e., $\frac{\sum_k|\zeta(\mathbf{F}^l;k)|}{\sum_k|S_c(\mathbf{F}^l)-S_c(\mathbf{F}^l\backslash\mathbf{F}^{lk})|}$ and $\left|\frac{\varepsilon(\mathbf{F}^l)}{S_c(\mathbf{F}^l)}\right|$) of 1000 input images in the last spatial layers of several classical CNN models, with the class of interest c set as the top-1 predicted class. We empirically found that the average of these terms are rather small for all the models as shown in Fig. 2. In contrast, these terms are rather large in shadow layers, a visualization of $\varepsilon(\mathbf{F}^l)$ in different layers of VGG16 model is provided in Appendix. Therefore, without considering $\zeta(\mathbf{F}^l;k)$[1] and $\varepsilon(\mathbf{F}^l)$, to minimize Eq. (6), we can calculate an approximate optimal solution α_c^k by making the terms in $|\cdot|$ equal to zero:

$$\alpha_c^k = \sum_{x,y}\left(\frac{F^{lk}(x,y)}{\sum_{x,y}F^{lk}(x,y)}\frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk}(x,y)}\right). \tag{7}$$

[1]To be precise, $\frac{\sum_k|\zeta(\mathbf{F}^l;k)|}{\sum_k|S_c(\mathbf{F}^l)-S_c(\mathbf{F}^l\backslash\mathbf{F}^{lk})|}$ is small can not indicate that $\frac{|\zeta(\mathbf{F}^l;k)|}{|S_c(\mathbf{F}^l)-S_c(\mathbf{F}^l\backslash\mathbf{F}^{lk})|}$ is small for all the feature maps. Exceptions exist indeed but they usually happen in the unimportant feature maps whose removing only lead to a tiny score change. For those feature maps, $\zeta(\mathbf{F}^l;k)$ is still rather small and can be ignored because it has little influence on the final visualization. A visual example is provided in Appendix.

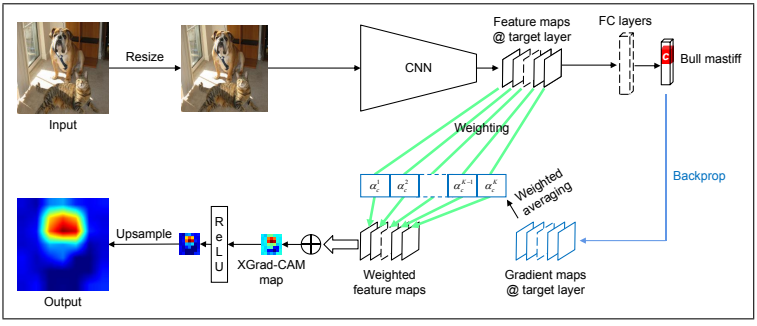


Figure 3: An overview of the XGrad-CAM scheme.

In this case, we define our XGrad-CAM by substituting w_c^k in Eq. (1) with α_c^k :

$$M_c^{\text{XGrad-CAM}}(x,y) = \sum_{k=1}^{K} \left(\alpha_c^k F^{lk}(x,y)\right), \tag{8}$$

similar to other CAM methods, by rectifying the resulting map using ReLU function and upsampling the map to the input size, we can finally identify the image regions responsible for the class c as shown in Fig. 3. Besides, we also propose Guided XGrad-CAM by multiplying the up-sampled XGrad-CAM by the Guided Backprop [28] element-wisely.

Note that XGrad-CAM is a generalization of CAM [33] since it is identical to CAM for the GAP-CNNs but can be applied to arbitrary CNN models (see Appendix for the proof).

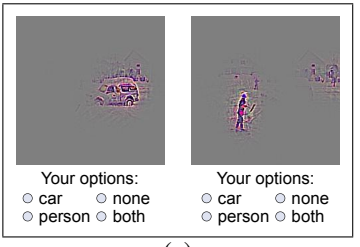
4 Experiments and Results

In this section, we mainly evaluate the performance of different CAM methods, including Grad-CAM [24], Grad-CAM++ [3], Ablation-CAM [2] and our XGrad-CAM. We argue that accurate localization of the objects of interest in an input image is necessary for an ideal visualization approach. Therefore, we evaluate the visualization quality from "class-discriminability" (see Sec. 4.2) and "localization capability" (see Sec. 4.3). In addition, we also analyze the rationality of existing methods from the perspective of axiom in Sec. 4.4.

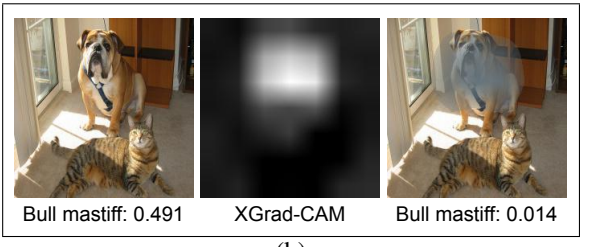
4.1 Experimental Setup

All of the existing methods are based on Eq. (1) but with different weights w_c^k, that is:

- Grad-CAM. The weight of each feature map is defined as $\frac{1}{Z}\sum_{x,y}\frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk}(x,y)}$ where Z is the number of units in the k-th feature map.
- Grad-CAM++. The weight of each feature map is defined as $\sum_{x,y} a_c^k(x,y)\text{ReLU}(\frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk}(x,y)})$ where $a_c^k(x,y)$ is a closed form weight based on an assumption that the class score is a linear function of feature maps.
- Ablation-CAM. The weight of each feature map is defined as $\frac{S_c(\mathbf{F}^l)-S_c(\mathbf{F}^l\backslash\mathbf{F}^{lk})}{\sum_{x,y}F^{lk}(x,y)}$, it removes the dependence of gradients. Note that the original weight of each feature map in Ablation-CAM [2] is defined as $\frac{S_c(\mathbf{F}^l)-S_c(\mathbf{F}^l\backslash\mathbf{F}^{lk})}{||\mathbf{F}^{lk}||}$. Since the selected target layer of



(a)



(b)

Figure 4: (a) A game of "What do you see" to evaluate the class-discriminability of each CAM method. Subject needs to answer what is being depicted in the visualization; (b) An example of XGrad-CAM visualization and its corresponding perturbed image.

Method	Class_discrimination	Confidence_drop	Efficiency (s)
Grad-CAM [24]	≈**0.709**	0.469	**0.021**
Grad-CAM++[3]	≈0.308	**0.494**	0.022
Ablation-CAM [4]	≈0.700	0.484	0.735
XGrad-CAM	≈0.702	0.491	**0.021**

Table 1: Results of class discrimination analysis and perturbation analysis. It is shown that Grad-CAM++ [3] is not class-discriminative, Grad-CAM [24] is not good enough in localizing the object of interest, Ablation-CAM [4] is time-consuming.

CAM methods is usually ReLU rectified, the responses of the feature maps are always positive, we set $||\mathbf{F}^{lk}||$ as $\sum_{x,y} F^{lk}(x,y)$ in this paper[2]. It is easy to verify that Ablation-CAM here totally satisfies the axiom of sensitivity.

These CAM methods are performed on the last spatial layer of VGG-16 model [25] pre-trained on the ImageNet. For GAP-CNNs (e.g., ResNet-101 [9] and Inception_V3 [31]), it can be proved that Grad-CAM, Ablation-CAM and XGrad-CAM achieve the same performance on the last spatial layers of the models (refer to Appendix for the detailed proof). All experiments are implemented in Pytorch [19] and conducted on an NVIDIA Titan Xp GPU.

4.2 Class Discrimination Analysis

A good visualization of CNN's decision should be class-discriminative. Specifically, the visualization method should only highlight the object belonging to the class of interest in an image when there are objects labeled with several different classes.

To evaluate the ability of class discrimination, we followed the subjective evaluation method used in [3]. Specifically, we first finetuned the pre-trained VGG-16 model on the Pascal VOC 2007 training set. The images in VOC set usually contain multiple objects belonging to different classes. We then selected 100 images from VOC 2007 validation set that contain exactly two classes. For each image and each CAM method, we used the guided version of the CAM method to generate a pair of class-specific visualizations as shown in Fig. 4 (a). These visualizations were then shown to 5 individuals who were asked to answer a choice question: what class is highlighted by the visualizations. Note that the options also include "none" and "both" which are both incorrect answers.

Quantitative results are shown in Table 1. The class-discriminative evaluation is subjective, but it is clear that the performance of Grad-CAM, Ablation-CAM and XGrad-CAM

[2] $||\mathbf{F}^{lk}||$ is roughly set to $S_c(\mathbf{F}^l)$ in the original paper.

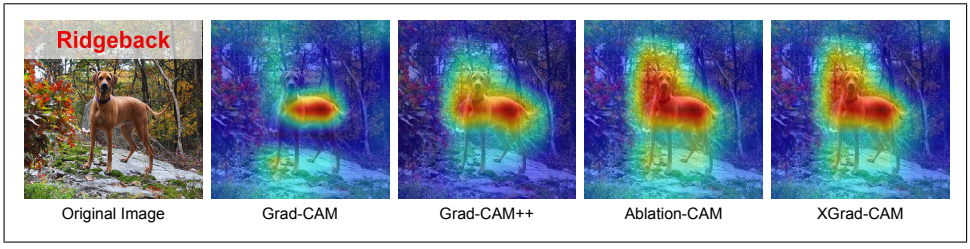


Figure 5: Example explanation maps generated by Grad-CAM [24], Grad-CAM++ [3], Ablation-CAM [4] and our XGrad-CAM.

are very similar. On the other hand, Grad-CAM++ performs much worse than the other three methods. For more visualization results and the reason why we evaluated the class-discriminability of different CAM methods using their guided versions rather than themselves, please refer to the Appendix.

4.3 Perturbation Analysis

The localization capability of CAM methods is usually evaluated by perturbation analysis [3, 4, 24]. The underlying assumption is that the perturbation of relevant regions in an input image should lead to a decrease in class confidence.

We followed the evaluation scheme used in [3, 4] for perturbation analysis. The experiments were conducted on the ILSVRC-12 validation set [22]. Take XGrad-CAM for example, each image $\mathbf{I}_i$ in the dataset was first fed to the VGG-16 model to predict its top-1 class. Then, XGrad-CAM method was used to generate a corresponding heatmap $\mathbf{H}_i$ for the predicted class. Inspired by the meaningful perturbation illustrated in [3], we perturbed the original image by masking the regions highlighted by the XGrad-CAM method:

$$\widetilde{\mathbf{I}}_i = \mathbf{I}_i \circ (1 - \mathbf{M}_i) + \mu \mathbf{M}_i, \tag{9}$$

where $\mathbf{M}_i$ is a mask based on the original heatmap $\mathbf{H}_i$. Specifically, in the mask, only the pixels corresponding to the top 20% value of the heatmap are set equal to the heatmap, while the rest are set to 0. "$\circ$" represents the element-wise multiplication and μ is the mean value used in the input normalization. A perturbed example is shown in Fig. 4 (b). It is shown that with the perturbation, the confidence of the target class "Bull mastiff" decreases sharply.

Then, we computed the difference of the class confidence (the output of the softmax layer) between the original image and the perturbed image:

$$Confidence\_drop = \frac{1}{N}\sum_{i=1}^{N}\frac{P_c(\mathbf{I}_i) - P_c(\widetilde{\mathbf{I}}_i)}{P_c(\mathbf{I}_i)}, \tag{10}$$

where $P_c(\mathbf{I}_i)$ and $P_c(\widetilde{\mathbf{I}}_i)$ are the class confidences of the original image $\mathbf{I}_i$ and perturbed image $\widetilde{\mathbf{I}}_i$, respectively, N is the total number of images in dataset. If the heatmap has highlighted the regions that are most important for class c, the confidence drop is expected to be larger.

The results are shown in Table 1, we can see that XGrad-CAM achieves better performance than Grad-CAM (0.491 v.s. 0.469). Ablation-CAM performs similar to XGrad-CAM, but it is much more time-consuming than XGrad-CAM (about 40 times), it has to run hundreds of forward propagation per image. While Grad-CAM++ achieves the best

Method	Sensitivity	Conservation
Grad-CAM [24]	0.313	0.303
Grad-CAM++[3]	$\gg 1$	$\gg 1$
Ablation-CAM [4]	**0**	0.145
XGrad-CAM	0.085	**0.051**

Table 2: Results of axiom analysis on the ILSVRC-12 validation set when applying CAM methods on the last spatial layers of VGG-16 model.

performance, its class-discriminability is lost (refer to Section 4.2). Note that, the class-discriminability cannot be reflected by the confidence drop on the ILSVRC-12 validation set because images in this dataset usually contain a single object class. Visual example results generated by the different CAM methods are shown in Fig. 5, it can be observed that the result achieved by XGrad-CAM covers a more complete area of the object than Grad-CAM.

To summarize, Grad-CAM++ [3] is not class-discriminative, Ablation-CAM [4] is time-consuming, Grad-CAM [24] is not good enough in localizing the object of interest. Therefore, considering the property of class discrimination, efficiency and the localization performance comprehensively, our XGrad-CAM is a promising visualization scheme in practice.

4.4 Axiom Analysis

To further study whether the existing CAM methods satisfy the axioms of sensitivity and conservation, we conduct axiom analysis on the ILSVRC-12 validation set. Specifically, the sensitivity of a general CAM method of Eq. (1) can be measured by:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{k=1}^{K}\left|S_c(\mathbf{F}_i^l)-S_c(\mathbf{F}_i^l\backslash\mathbf{F}_i^{lk})-\sum_{x,y}\left(w_c^k F_i^{lk}(x,y)\right)\right|}{\sum_{k=1}^{K}\left|S_c(\mathbf{F}_i^l)-S_c(\mathbf{F}_i^l\backslash\mathbf{F}_i^{lk})\right|}. \tag{11}$$

where $\mathbf{F}_i^l$ is the response of the target layer of the i-th image in the dataset, c is the top-1 class predicted by the VGG-16 model. Analogously, the conservation can be measured by:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\left|S_c(\mathbf{F}_i^l)-\sum_{k=1}^{K}\sum_{x,y}\left(w_c^k F_i^{lk}(x,y)\right)\right|}{\left|S_c(\mathbf{F}_i^l)\right|}, \tag{12}$$

We report the comparison results of the different CAM methods in Table 2. Note that, the lower value of the sensitivity and conservation indicates that the method suits the axioms better. It is clear that the Grad-CAM++ breaks the axioms of sensitivity and conservation with poor performance in the axiomatic evaluation. This may further explain why Grad-CAM++ cannot achieve comparable performance in class discrimination analysis. The results imply that it may be important to consider the axioms in designing visualization methods.

5 Conclusion

In this paper, we present a novel visualization method called XGrad-CAM motivated by the axioms of sensitivity and conservation. A clear mathematical explanation is provided to fill the gap in interpretability for CAM visualization methods. Experimental results show that our XGrad-CAM enhances Grad-CAM in terms of sensitivity and conservation, and significantly improves the visualization performance compared with Grad-CAM. We also give a reasonable explanation why existing methods (i.e., Grad-CAM and Ablation-CAM) can be effective from the perspective of axioms.

6 Acknowledgement

This work was partially supported by the National Natural Science Foundation of China (No. 61972435) and the China Scholarship Council (CSC).

References

[1] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, pages 3319–3327, 2017.

[3] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, pages 839–847, 2018.

[4] Saurabh Desai and Harish G Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *WACV*, pages 972–980, 2020.

[5] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*, pages 7714–7722, 2019.

[6] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, pages 3429–3437, 2017.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[8] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys*, 51(5):1–42, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[11] Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Sven Dähne. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*, 2016.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.

[13] Juncheng Li, Frank Schmidt, and Zico Kolter. Adversarial camera stickers: A physical camera-based attack on deep learning systems. In *ICML*, pages 3896–3904, 2019.

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[15] Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3): 233–255, 2016.

[16] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

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

[18] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*, 2019.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019.

[20] Zhuwei Qin, Fuxun Yu, Chenchen Liu, and Xiang Chen. How convolutional neural networks see the world—a survey of convolutional neural network visualization methods. *Mathematical Foundations of Computing*, 1(2):149–180, 2018.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015.

[23] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

[24] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.

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

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

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

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

[29] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

[30] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, pages 3319–3328, 2017.

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[32] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.

[33] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016.

[34] Keyang Zhou and Bernhard Kainz. Efficient image evidence analysis of cnn classification results. *arXiv preprint arXiv:1801.01693*, 2018.

[35] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

A Proof

In this section, we aim to demonstrate that given an arbitrary layer in ReLU-CNNs, for any class of interest, there exists a specific equation between the class score and the feature maps of the layer.

For a ReLU-CNN which only has ReLU rectification as its nonlinearity, the following equation holds for an arbitrary layer l:

$$u_j^{l+1} = \sum_i \left(\frac{\partial u_j^{l+1}}{\partial u_i^l} u_i^l \right) + b_j^{l+1}, \tag{13}$$

where u_i^l represents an unit in layer l, u_j^{l+1} represents an unit in layer $l+1$, $\frac{\partial u_j^{l+1}}{\partial u_i^l}$ is the gradient of u_j^{l+1} w.r.t. u_i^l, b_j^{l+1} is the bias term asscociated with the unit u_j^{l+1}. Note that, if unit u_j^t is an output of a ReLU or pooling layer, the corresponding bias term b_j^t is zero.

We then prove our statement (i.e., Eq.(5) in the main paper) using *mathematical induction* [1]. In the top layer L, the response of the c-th unit is exactly the class score of interest S_c in the main paper, and it is easy to verify that:

$$u_c^L = \sum_i \left(\frac{\partial u_c^L}{\partial u_i^{L-1}} u_i^{L-1} \right) + b_c^L, \tag{14}$$

Suppose that for layer l ($l < L$):

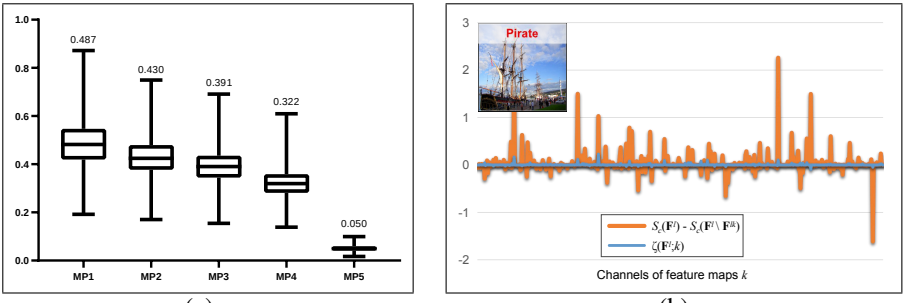
$$u_c^L = \sum_i \left(\frac{\partial u_c^L}{\partial u_i^l} u_i^l \right) + \sum_{t=l+1}^{L} \sum_k \frac{\partial u_c^L}{\partial u_k^t} b_k^t, \tag{15}$$

Then, for layer $l-1$, it holds:

$$\begin{aligned}
&\sum_{i'} \left(\frac{\partial u_c^L}{\partial u_{i'}^{l-1}} u_{i'}^{l-1} \right) + \sum_{t=l}^{L} \sum_{k'} \frac{\partial u_c^L}{\partial u_{k'}^t} b_{k'}^t \\
&= \sum_{i'} \left(\frac{\partial u_c^L}{\partial u_{i'}^{l-1}} u_{i'}^{l-1} \right) + \sum_{k'} \frac{\partial u_c^L}{\partial u_{k'}^l} b_{k'}^l + \sum_{t=l+1}^{L} \sum_k \frac{\partial u_c^L}{\partial u_k^t} b_k^t \\
&= \sum_{i'} \left(\sum_i \left(\frac{\partial u_c^L}{\partial u_i^l} \frac{\partial u_i^l}{\partial u_{i'}^{l-1}} \right) u_{i'}^{l-1} \right) + \sum_{k'} \frac{\partial u_c^L}{\partial u_{k'}^l} b_{k'}^l + \sum_{t=l+1}^{L} \sum_k \frac{\partial u_c^L}{\partial u_k^t} b_k^t \\
&= \sum_i \frac{\partial u_c^L}{\partial u_i^l} \left(\sum_{i'} \left(\frac{\partial u_i^l}{\partial u_{i'}^{l-1}} u_{i'}^{l-1} \right) \right) + \sum_{k'} \frac{\partial u_c^L}{\partial u_{k'}^l} b_{k'}^l + \sum_{t=l+1}^{L} \sum_k \frac{\partial u_c^L}{\partial u_k^t} b_k^t \\
&= \sum_i \frac{\partial u_c^L}{\partial u_i^l} \left(\sum_{i'} \left(\frac{\partial u_i^l}{\partial u_{i'}^{l-1}} u_{i'}^{l-1} \right) + b_i^l \right) + \sum_{t=l+1}^{L} \sum_k \frac{\partial u_c^L}{\partial u_k^t} b_k^t \\
&= \sum_i \left(\frac{\partial u_c^L}{\partial u_i^l} u_i^l \right) + \sum_{t=l+1}^{L} \sum_k \frac{\partial u_c^L}{\partial u_k^t} b_k^t
\end{aligned} \tag{16}$$

i.e.,

$$u_c^L = \sum_{i'} \left(\frac{\partial u_c^L}{\partial u_{i'}^{l-1}} u_{i'}^{l-1} \right) + \sum_{t=l}^{L} \sum_{k'} \frac{\partial u_c^L}{\partial u_{k'}^t} b_{k'}^t, \tag{17}$$



(a) (b)

Figure 6: (a) Normalized $\varepsilon(\mathbf{F}^l)$ in different layers of VGG-16 model. "MP" represents for Maxpooling layer. The mean values are provided above the box-plots; (b) $\frac{|\zeta(\mathbf{F}^l;k)|}{|S_c(\mathbf{F}^l)-S_c(\mathbf{F}^l\setminus\mathbf{F}^{lk})|}$ is small for most of the feature maps. Exceptions usually happen in the unimportant feature maps.

This means that for an arbitrary layer, the class score equals to the sum of gradient$\times$feature plus an extra bias term.

B $\varepsilon(\mathbf{F}^l)$ and $\zeta(\mathbf{F}^l;k)$

$\varepsilon(\mathbf{F}^l)$ is the bias term in Eq. (5) in the main paper. We calculated $\left|\frac{\varepsilon(\mathbf{F}^l)}{S_c(\mathbf{F}^l)}\right|$ of 1000 input images in different layers of VGG-16 model, with the class of interest c set as the top-1 predicted class. Fig. 6(a) shows that this term is rather large in shallow layers.

$\zeta(\mathbf{F}^l;k)$ is a bias term in Eq. (6) in the main paper. Given an input example, Fig. 6(b) shows the values of $S_c(\mathbf{F}^l) - S_c(\mathbf{F}^l\setminus\mathbf{F}^{lk})$ and $\zeta(\mathbf{F}^l;k)$ w.r.t. all the feature maps in the last spatial layer of VGG16 model. It can be seen that $\frac{|\zeta(\mathbf{F}^l;k)|}{|S_c(\mathbf{F}^l)-S_c(\mathbf{F}^l\setminus\mathbf{F}^{lk})|}$ is rather small for most of the feature maps. Exceptions usually happen in the unimportant feature maps whose removing only lead to a tiny score change.

C CAM, Grad-CAM, Ablation-CAM and XGrad-CAM on GAP-CNNs

In this section, we prove that for GAP-CNNs (e.g., ResNet-101, Inception_v3), CAM [33], Grad-CAM [24], Ablation-CAM [4] and our XGrad-CAM achieve the same performance on the last spatial layers of the models.

GAP-CNNs usually consist of fully-convolution layers, global average pooling and a linear classifier with softmax. Specifically, let $\mathbf{F}^l$ be the last spatial layer, the output of the global average pooling is:

$$A^k = \frac{1}{Z}\sum_{x,y} F^{lk}(x,y) \tag{18}$$

where Z is the number of units in the k-th feature map. The score of class c is exactly a

weighted sum of A^k since the classifier is linear:

$$S_c = \sum_{k=1}^{K} \left(w_c^k A^k \right) + b_c \tag{19}$$

where w_c^k is the weight connecting the k-th feature map with the c-th class, b_c is a bias. Combining Eq. (18) and Eq. (19), we have:

$$S_c(\mathbf{F}^l) = \frac{1}{Z} \sum_{x,y} \sum_{k=1}^{K} \left(w_c^k F^{lk}(x,y) \right) + b_c \tag{20}$$

The weight of CAM [33] is then defined as w_c^k.

For a GAP-CNN, we can simply get that $\forall x,y$, $\frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk}(x,y)} = \frac{1}{Z} w_c^k$ using the Chain Rule. Recall the definition of the weights in Grad-CAM [24], we have:

$$\frac{1}{Z} \sum_{x,y} \frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk}(x,y)} = \frac{1}{Z} w_c^k. \tag{21}$$

Recall the definition of the weights in Ablation-CAM [4], we have:

$$\frac{S_c(\mathbf{F}^l) - S_c(\mathbf{F}^l \backslash \mathbf{F}^{lk})}{\sum_{x,y} F^{lk}(x,y)} = \frac{w_c^k A^k}{Z A^k} = \frac{1}{Z} w_c^k. \tag{22}$$

Recall the definition of the weights in XGrad-CAM, we have:

$$\sum_{x,y} \left(\frac{F^{lk}(x,y)}{\sum_{x,y} F^{lk}(x,y)} \frac{\partial S_c(\mathbf{F}^l)}{\partial F^{lk}(x,y)} \right) = \sum_{x,y} \left(\frac{F^{lk}(x,y)}{\sum_{x,y} F^{lk}(x,y)} \frac{1}{Z} w_c^k \right) = \frac{1}{Z} w_c^k. \tag{23}$$

It shows that the weights of Grad-CAM [24], Ablation-CAM [4] and XGrad-CAM are exactly the same in the case of GAP-CNNs. Besides, they are also identical to the weight of CAM [33] expect a constant Z, which makes no difference for visualization. Therefore, we can conclude that CAM [33], Grad-CAM [24], Ablation-CAM [4] and XGrad-CAM achieve the same performance on the last spatial layers of GAP-CNNs.

D Additional Visualization Results

In the section of class discrimination analysis in the main paper, we evaluated the class-discriminability of different CAM methods using their guided versions rather than themselves. The motivation comes from two aspects. First, the guided versions have the same class-discriminability as the original versions. As shown in Fig. 7, we visualized several visualization results of XGrad-CAM, Guided Backprop [28] and Guided XGrad-CAM. It is shown that Guided XGrad-CAM inherits the class-discriminability of XGrad-CAM completely. This phenomenon applies to all the other CAM methods. Second, the results of guided versions provides a better visualization for the objects of interest with more object details. It helps the subjects make their decisions more accurately and efficiently in the game of "What do you see" as shown in Fig.4(a) in the main paper.

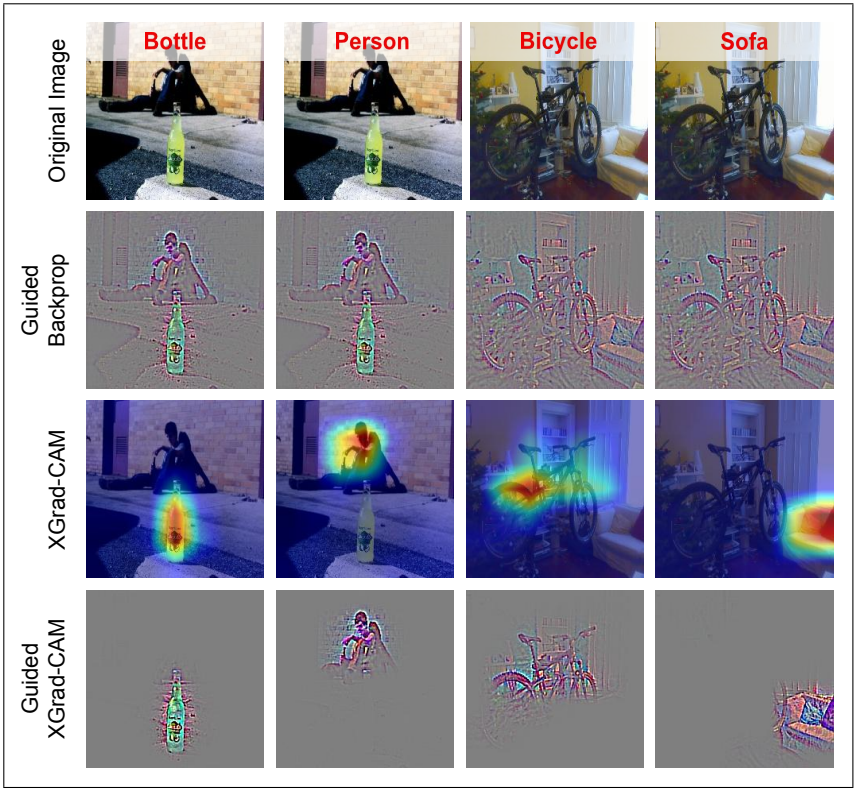


Figure 7: Several visualization results of XGrad-CAM, Guided Backprop and Guided XGrad-CAM.

Fig. 8 presents several qualitative results in VOC 2007 validation set to further compare the class-discriminability of different CAM methods. We can see that if there are objects belonging to multiple classes in an image, Grad-CAM++ also highlights regions of irrelevant classes. Clearly, Grad-CAM++ is not class-discriminative compared with the other three CAM methods.

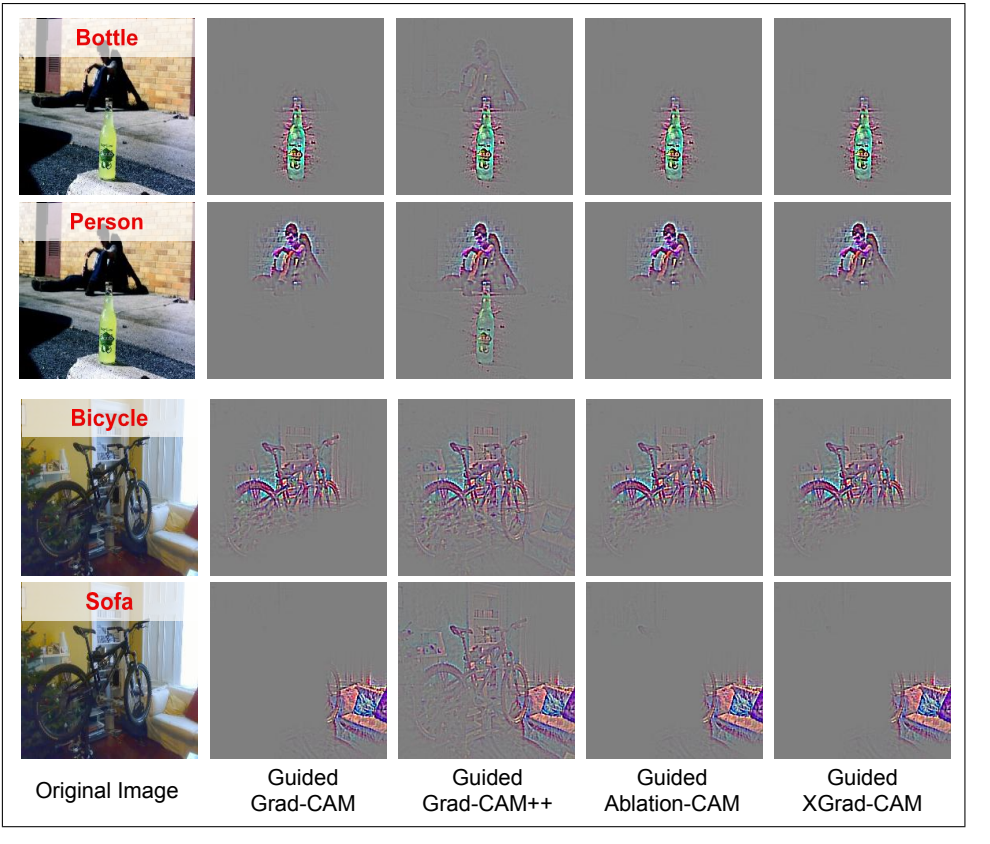


Figure 8: Additional visualization results to compare the class-discriminability of different CAM methods.