# LIFT-CAM: Towards Better Explanations for Class Activation Mapping

Hyungsik Jung Youngrock Oh
AI Vision Lab., Samsung SDS
Seoul 06765, South Korea
{hs89.jung, y52.oh}@samsung.com

## Abstract

*Increasing demands for understanding the internal behaviors of convolutional neural networks (CNNs) have led to remarkable improvements in explanation methods. Particularly, several class activation mapping (CAM) based methods, which generate visual explanation maps by a linear combination of activation maps from CNNs, have been proposed. However, the majority of the methods lack a theoretical basis in how to assign their weighted linear coefficients. In this paper, we revisit the intrinsic linearity of CAM w.r.t. the activation maps. Focusing on the linearity, we construct an explanation model as a linear function of binary variables which denote the existence of the corresponding activation maps. With this approach, the explanation model can be determined by the class of additive feature attribution methods which adopts SHAP values as a unified measure of feature importance. We then demonstrate the efficacy of the SHAP values as the weight coefficients for CAM. However, the exact SHAP values are incalculable. Hence, we introduce an efficient approximation method, referred to as LIFT-CAM. On the basis of DeepLIFT, our proposed method can estimate the true SHAP values quickly and accurately. Furthermore, it achieves better performances than the other previous CAM-based methods in qualitative and quantitative aspects.*

## 1. Introduction

Recently, convolutional neural networks (CNNs) have achieved excellent performance in various real-world vision tasks. However, it is difficult to explain their predictions due to a lack of understanding of their internal behavior. To grasp why a model makes a certain decision, numerous *saliency methods* have been proposed. The methods generate *visual explanation maps* that represent, by interpreting CNNs, pixel-level importances w.r.t. which regions in an input image are responsible for the model's decision and which parts are not. Towards better comprehension of CNNs, *class activation mapping (CAM)* based methods, which utilize the responses of a convolutional layer for explanations, have been widely adopted for saliency methods.

CAM-based methods [21, 15, 4, 19, 5, 8] (abbreviated as CAMs in the remainder of this paper) *linearly* combine activation maps to produce visual explanation maps. Since the activation maps are fixed for a given pair of an input image and a model, the weight coefficients for a linear combination govern the performance of the methods. Therefore, a reasonable method of regulating the coefficients is all we need. However, the majority of CAMs rely on heuristic conjectures when deciding their weight coefficients in the absence of a theoretical background. Specifically, the underlying linearity of CAM w.r.t. the activation maps is not fully considered in the methods. In addition, a set of desirable conditions which are expected to be satisfied in a trustworthy explanation model are disregarded.

In this work, we leverage the linearity of CAM to analytically determine the coefficients beyond heuristics. Focusing on the fact that CAM defines an explanation map using a linear combination of activation maps, we consider an explanation model as a linear function of the binary variables which represent the existence of the associated activation maps. By this assumption, each activation map can be deemed as an individual feature in the class of *additive feature attribution methods* [13, 3, 17, 10], which adopts *SHAP values* [11] as a unique solution, with three desirable conditions (described in Sec 2.2). Consequently, we can decide the weight coefficients as the corresponding SHAP values. However, the exact SHAP values are uncomputable. Thus, we propose a novel saliency method, *LIFT-CAM*, which efficiently approximates the SHAP values of the activation maps using DeepLIFT [17]. Our contributions can be summarized as follows:

- We suggest a novel framework for reorganizing the problem of finding out the plausible visual explanation map of CAM to that of determining the reliable solution for the explanation model using the additive feature attribution methods. The recently proposed Ablation-CAM [5] and XGrad-CAM [8] can be reinterpreted by this framework.

- We formulate the SHAP values of the activation maps as a unified solution for the suggested framework. Furthermore, we verify the excellence of the values.
- We introduce a new saliency method, LIFT-CAM, employing DeepLIFT. The method can estimate the SHAP values of the activation maps precisely. It outperforms previous CAMs qualitatively and quantitatively by providing reliable visual explanations with a single backward propagation.

## 2. Related works

### 2.1. CAM-based methods

**Visual explanation map.** Let $f$ be an original prediction model and $c$ denote a target class of interest. CAMs [21, 15, 4, 19, 5, 8] aim to explain the target output of the model for a specific input image $x$ (i.e., $f^c(x)$) through the visual explanation map, which can be generated by:

$$L^c_{CAM}(\mathbf{A}) = ReLU(\sum_{k=1}^{N_l} \alpha_k A_k) \tag{1}$$

with $\mathbf{A} = f^{[l]}(x)$, where $f^{[l]}$ denotes the output of the $l$-th layer. $A_k$ is a $k$-th activation map of $\mathbf{A}$ and $\alpha_k$ is a corresponding weight coefficient (i.e., an importance) of $A_k$, respectively. $N_l$ indicates the number of activation maps for layer $l$. This concept of linearly combining activation maps was firstly proposed by [21], leading to its variants.

**Previous methods.** Grad-CAM [15] decides the coefficient of a specific activation map by averaging the gradients over all activation neurons in that map. Grad-CAM++ [4], which is a modified version of Grad-CAM, focuses on positive influences of neurons considering higher-order derivatives. However, the gradients of deep neural networks tend to diminish due to the gradient saturation problem. Hence, using the unmodified raw gradients induces failure of localization for relevant regions.

To overcome this limitation, gradient-free CAMs have been proposed. Score-CAM [19] overlaps normalized activation maps to an input image and makes predictions to acquire the coefficients. Ablation-CAM [5] defines a coefficient as the fraction of decline in the target output when the associated activation map is removed. The two CAMs are free from the saturation issues, but they are time-consuming because they require $N_l$ forward propagations to acquire the coefficients.

All the methods described above evaluate their coefficients in a heuristic way. XGrad-CAM [8] addresses this issue by suggesting two axioms. The authors mathematically proved that the coefficients satisfying the axioms converge to the summations of the "input×gradient" values of the activation neurons. However, their derivation is demonstrated only for ReLU-CNN.

### 2.2. SHapley Additive exPlanations

**Additive feature attribution method.** SHAP [11] is a unified explanation framework for the class of additive feature attribution methods. The additive feature attribution method follows:

$$g(z^{'}) = \phi_0 + \sum_{i=1}^{M} \phi_i z_i^{'} \tag{2}$$

where $g$ is an explanation model of an original prediction model $f$ and $M$ is the number of features. $\phi_i$ denotes an importance of the $i$-th feature and $\phi_0$ is a baseline explanation. $z^{'} \in \{0,1\}^M$ indicates a binary vector in which each entry represents the existence of the corresponding original feature; 1 for *presence* and 0 for *absence*. The methods are designed to ensure $g(z^{'}) \approx f(h_x(z^{'}))$ whenever $z^{'} \approx x^{'}$, with a mapping function $h_x$ which satisfies $x = h_x(x^{'})$. While several existing attribution methods [13, 3, 17, 10] match Eq. (2), only one explanation model in the class satisfies the three desirable properties: *local accuracy*, *missingness*, and *consistency* [11].

**SHAP values.** A feature attribution of the explanation model which obeys Eq. (2) while adhering to the above three properties is defined as SHAP values [11] and can be formulated by:

$$\phi_i = \sum_{z^{'} \subset x^{'}} \frac{(M - |z^{'}| - 1)!|z^{'}|!}{M!}[f(h_x(z^{'})) - f(h_x(z^{'} \backslash i))] \tag{3}$$

where $|z^{'}|$ denotes the number of non-zero entries of $z^{'}$ and $z^{'} \subset x^{'}$ represents all $z^{'}$ vectors, where the non-zero entries are a subset of the non-zero entries in $x^{'}$. In addition, $z^{'} \backslash i$ indicates $z_i^{'} = 0$. This definition of the SHAP values intimately aligns with the classic Shapley values [16].

### 2.3. DeepLIFT

DeepLIFT [17] focuses on the difference between an *original* activation and a *reference* activation. It propagates the difference through a network to assign the contribution score to each input neuron by linearizing non-linear components in the network. Through this technique, the gradient saturation problem is stabilized.

Let $o$ represent the output of the target neuron and $x = (x_1, \ldots, x_n)$ be input neurons whose reference values are $r = (r_1, \ldots, r_n)$. The contribution score of the $i$-th input feature $C_{\Delta x_i \Delta o}$ quantifies the influence of $\Delta x_i = x_i - r_i$ on $\Delta o = f(x) - f(r)$. In addition, DeepLIFT satisfies the *summation-to-delta* property as below:

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta o} = \Delta o. \tag{4}$$

Note that if we set $C_{\Delta x_i \Delta o} = \phi_i$ and $f(r) = \phi_0$, then Eq. (4) matches Eq. (2). Therefore, DeepLIFT is also an additive feature attribution method for which the contribution

scores are among the solutions for Eq. (2). It *approximates* the SHAP values efficiently, satisfying the local accuracy and missingness [11].

## 3. Proposed Methodology

In this section, we clarify the problem formulation of CAM and suggest an approach to resolve it analytically. First, we suggest a framework that defines a linear explanation model and decides the weight coefficients of CAM based on that model. Then, we formulate the SHAP values of the activation maps as a unified solution for the model. Finally, we introduce a fast and precise approximation method for the SHAP values using DeepLIFT.

### 3.1. Problem formulation of CAM

As identified in Eq. (1), CAM produces a visual explanation map $L^c_{CAM}$ linearly w.r.t. the activation maps $\{A_k\}_{k\in\{1,...,N^l\}}$ except for ReLU, which is applied for the purpose of only considering the positive influence on the target class $c$. In addition, the matrix of the complete activation maps $\mathbf{A} = \{A_k\}_{k\in\{1,...,N^l\}}$ does not change for a given pair of the input image $x$ and the model $f$. Thus, the responsibility of $L^c_{CAM}$ is controlled by the weight coefficients $\{\alpha_k\}_{k\in\{1,...,N^l\}}$, which represent the importance scores of the associated activation maps. To summarize, the purpose of CAM is to find the optimal coefficients $\{\alpha^{opt}_k\}_{k\in\{1,...,N^l\}}$ for a linear combination in order to generate $L^c_{CAM}$, which can reliably explain the target output $f^c(x)$.

### 3.2. The suggested framework

*How can we acquire the desirable coefficients in an analytic way?* To this end, we first consider each activation map as an individual feature (i.e., we have $N_l$ features) and define a binary vector $a' \in \{0,1\}^{N_l}$ of the features. In the vector, an entry $a'_k$ of 1 indicates that the corresponding activation map $A_k$ maintains its original activation values, and 0 means that it loses the values. Next, we specify an explanation model of CAM $g_{CAM}$ to interpret $f^c(x)$.

Since the explanation map of CAM $L^c_{CAM}$ is linear w.r.t. the activation maps $\mathbf{A}$ by definition, it is rational to assume that the explanation model $g_{CAM}$ is also linear w.r.t. the binary variables of the activation maps $a'$ as follows:

$$g_{CAM}(a') = \alpha_0 + \sum_{k=1}^{N_l} \alpha_k a'_k. \tag{5}$$

This linear explanation model can be solved by the class of additive feature attribution methods, matching Eq. (2) perfectly. Therefore, the SHAP values can be adopted as a *unified* solution for $\{\alpha_k\}_{k\in\{1,...,N^l\}}$ (see the supplementary materials for comparison with LIME [13]). Figure 1 represents the framework which is described in this section.

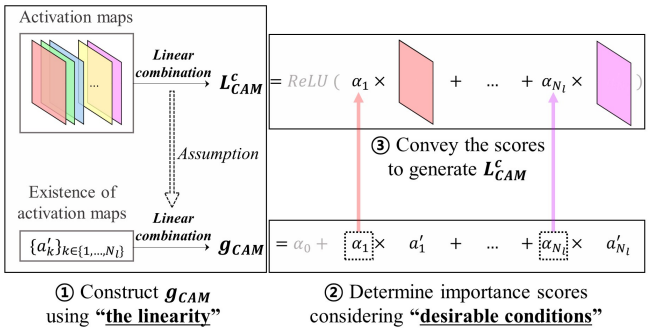


Figure 1. The suggested framework for determining the weight coefficients for CAM. First, we build a linear explanation model. Next, we determine the importance scores of activation maps by optimizing the explanation model based on additive feature attribution methods. Last, we convey the scores as the coefficients for CAM.

### 3.3. SHAP values of activation maps

Next, we define the SHAP values of activation maps. Let $F$ be a latter part of the original model $f$, from layer $l+1$ to layer $L-1$[1], where $L$ represents the total number of layers in $f$. Namely, we have $F(\mathbf{A}) = f^{[L-1]}(x)$. Additionally, we define a mapping function $h_{\mathbf{A}}$ that converts $a'$ into the embedding space of the activation maps: it satisfies $\mathbf{A} = h_{\mathbf{A}}(\mathbf{A}')$, where $\mathbf{A}'$ is a vector of ones. Specifically, $a'_k = 1$ is mapped to $A_k$ and $a'_k = 0$ to $\mathbf{0}$, which has the same dimension as $A_k$. Note that this is reasonable because an activation map exerts no influence on visual explanation when it has values of 0 for all activation neurons in Eq. (1).

Now, the SHAP values of the activation maps w.r.t. the class $c$ can be formulated by [11]:

$$\alpha^{sh}_k = \sum_{a'\subset\mathbf{A}'} \frac{(N_l - |a'| - 1)!|a'|!}{N_l!}[F^c(h_{\mathbf{A}}(a')) - F^c(h_{\mathbf{A}}(a'\backslash k))] \tag{6}$$

where $\alpha^{sh}_k$ is the SHAP value of the $k$-th activation map. The above equation implies that $\alpha^{sh}_k$ can be obtained by averaging marginal prediction differences between *presence* and *absence* of $A_k$ across all possible feature orderings. In this context, Algorithm 1 shows the overall procedure to approximate the SHAP values from a set of sampled orderings. We refer to the approximation from $|\Pi|$ orderings as SHAP-CAM$_{|\Pi|}$ throughout the paper. The higher $|\Pi|$, the closer SHAP-CAM$_{|\Pi|}$ approximates $\{\alpha^{sh}_k\}_{k\in\{1,...,N^l\}}$. We validate the effectiveness and the superiority of these SHAP attributions in our experimental section.

[1] It represents the logit layer which precedes the softmax layer.

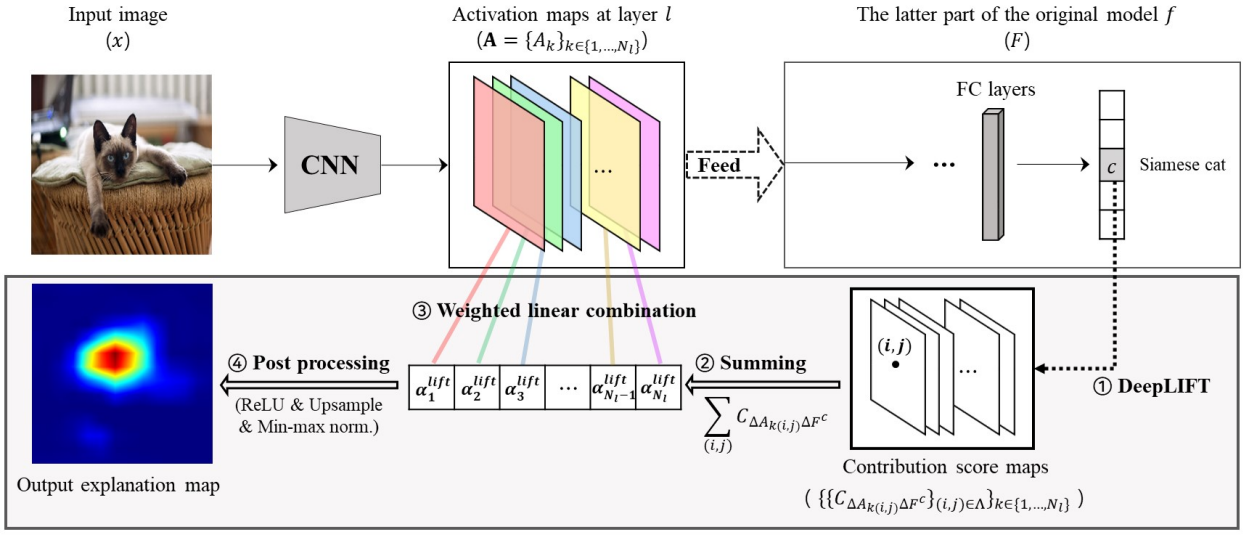


Figure 2. An overview of LIFT-CAM. First, we conduct DeepLIFT from the target output towards the activation maps and acquire the contribution score maps, for which each pixel presents $C_{\Delta A_{k(i,j)}\Delta F^c}$. Next, we quantify the importance of each activation map by summing all the contribution scores of itself. Then, we perform a linear combination of $\{\alpha_k^{lift}\}_{k\in\{1,\dots,N_l\}}$ and $\{A_k\}_{k\in\{1,\dots,N_l\}}$. Finally, we rectify the resulting map, upsample the map to the original image dimension, and normalize the map using the min-max normalization function.

**Algorithm 1** SHAP-CAM$_{|\Pi|}$ approximation

**Input** : $F$, $c$, $h_{\mathbf{A}}$, and a set of orderings $\Pi$ of $\{1, \cdots, N_l\}$
**Output:** $\{\alpha_k\}_{k\in\{1,\dots,N_l\}}$
**Initialize:** $\{\alpha_k\}_{k\in\{1,\dots,N_l\}} \leftarrow \mathbf{0}$
**for** each ordering $\pi$ in $\Pi$ **do**
  $a^{'} \leftarrow \mathbf{0}$
  **for** $i = 1, ..., N_l$ **do**
    $a^{'}_{\pi(i)} \leftarrow 1$
    $\alpha_{\pi(i)} \leftarrow \alpha_{\pi(i)} + F^c(h_{\mathbf{A}}(a^{'})) - F^c(h_{\mathbf{A}}(a^{'} \backslash \pi(i)))$
  **end**
**end**
$\{\alpha_k\}_{k\in\{1,\dots,N_l\}} \leftarrow \{\alpha_k\}_{k\in\{1,\dots,N_l\}}/|\Pi|$

### 3.4. Efficient approximation: LIFT-CAM

Through the experiment, we prove that a favorable visual explanation map can be achieved by the SHAP values. However, calculating the exact SHAP values is almost impossible. Therefore, we should consider an approximation approach. In this study, we propose a novel method, LIFT-CAM, that efficiently approximates the SHAP values of the activation maps using DeepLIFT[2] [17] (see the supplementary materials for analysis of other approximation algorithms).

First, we calculate the contribution score for every activation neuron through a single backward pass. Considering the *summation-to-delta* property of DeepLIFT, we define the contribution score of the specific activation map as the summation of the contribution scores of all neurons in that activation map, as follows:

$$\alpha_k^{lift} = C_{\Delta A_k \Delta F^c} = \sum_{(i,j)\in\Lambda} C_{\Delta A_{k(i,j)}\Delta F^c} \tag{7}$$

where $\Lambda = \{1, \dots, H\} \times \{1, \dots, W\}$ is a discrete activation dimension and $A_{k(i,j)}$ is an activation value at the $(i, j)$ location of $A_k$. Note that $\Delta$ denotes the difference-from-reference and the reference values (i.e., uninformative values) of all activation neurons are set to 0, aligning with SHAP. By this definition, $\{\alpha_k^{lift}\}_{k\in\{1,\dots,N^l\}}$ becomes a reliable solution for Eq. (5) while satisfying the local accuracy[3] and the missingness as below:

$$\sum_{k=1}^{N_l} \alpha_k^{lift} = F^c(\mathbf{A}) - F^c(\mathbf{0}). \tag{8}$$

Using these DeepLIFT attributions, LIFT-CAM can estimate the SHAP values of the activation maps with a single backward pass while alleviating the gradient saturation problem. Figure 2 shows an overview of our suggested LIFT-CAM. Additionally, the following two rationales motivate us to employ DeepLIFT for this problem.

[2]DeepLIFT-Rescale is adopted for approximation because the method can be easily implemented by overriding gradient operators. This convenience enables LIFT-CAM to be easily applied to a large variety of tasks.

[3]$\sum_{k=1}^{N_l} \alpha_k^{sh} = F^c(\mathbf{A}) - F^c(\mathbf{0})$.

1. DeepLIFT linearizes non-linear components to estimate the SHAP values. Therefore, DeepLIFT attributions tend to deviate from the true SHAP values in the case of passing through many overlapped non-linear layers during back-propagation (see the supplementary materials). However, for CAM, only the non-linearities in $F$ matter. Since CAM usually adopts the outputs of the last convolutional layer as its activation maps, almost all $F$ of state-of-the-art architectures contain few non-linearities (e.g., the VGG family), or are even fully linear (e.g., the ResNet family). Thus, the SHAP values of the activation maps can be approximated quite precisely (i.e., $\alpha^{lift} \approx \alpha^{sh}$) by DeepLIFT.

2. The reference value (i.e., the value of the *absent* feature) should be defined as an uninformative value. For this problem, the reference values of all activation neurons are definitely 0 because the value cannot influence $L^c_{CAM}$. This removes the need to heuristically choose the reference values and enables acquisition of the exact SHAP values using LIFT-CAM (i.e., $\alpha^{lift} = \alpha^{sh}$) for the architectures with linear $F$.

### 3.5. Rethinking Ablation-CAM and XGrad-CAM

The recently proposed Ablation-CAM [5] and XGrad-CAM [8] can be reinterpreted by our framework. Ablation-CAM defines the coefficients as below:

$$\alpha^{ab}_k = F^c(h(\mathbf{A}^{'})) - F^c(h(\mathbf{A}^{'} \backslash k)). \tag{9}$$

By adopting this specific marginal change as the coefficient, Ablation-CAM can be deemed as another approximation method for the SHAP values. Meanwhile, the coefficients of XGrad-CAM can be achieved by:

$$\alpha^{xg}_k = \sum_{(i,j)\in\Lambda} A_{k(i,j)} \times \frac{\partial F^c(\mathbf{A})}{\partial A_{k(i,j)}}. \tag{10}$$

[1] proved that this "input×gradient" attribution is equivalent to the relevance score of layer-wise relevance propagation (LRP) [3] for ReLU-CNN. Therefore, $\alpha^{xg}_k$ can be viewed as summing the relevance scores of the activation neurons in the $k$-th activation map, similar to the approach in LIFT-CAM. LRP is also an additive feature attribution method, and these LRP attributions $\{\alpha^{xg}_k\}_{k\in\{1,\dots,N^l\}}$ estimate the SHAP values as a solution for Eq. (5). However, both methods do not satisfy the local accuracy of SHAP (i.e., $\sum_{k=1}^{N_l} \alpha_k \neq \sum_{k=1}^{N_l} \alpha^{sh}_k$). This mismatch leads to less precise approximations compared to LIFT-CAM, resulting in less reliable explanations.

## 4. Experiments and Results

We now describe our experiments and show the results. We first validate the efficacy and the excellence of the SHAP values as the coefficients for CAM in Sec. 4.1. Then, we demonstrate how closely LIFT-CAM can estimate the SHAP values in Sec. 4.2. These two experiments provide justification to opt for LIFT-CAM as a responsible method of determining the coefficients for CAM. We then evaluate the performance of LIFT-CAM on the object recognition task in the context of image classification, comparing it with current state-of-the-art CAMs: Grad-CAM, Grad-CAM++, Score-CAM, Ablation-CAM, and XGrad-CAM in Sec. 4.3. Finally, we apply LIFT-CAM to the visual question answering (VQA) task in Sec. 4.4 to check the scalability of the method.

For all experiments except VQA, we employ the public classification datasets ImageNet [14] (ILSVRC 2012 validation set), PASCAL VOC [6] (2007 test set), and COCO [9] (2014 validation set). In addition, the VGG16 network trained on each dataset is analyzed for the experiments (see the supplementary materials for the experiments of ResNet50). We refer to the pretrained models from the torchvision[4] package for ImageNet and the TorchRay package[5] for VOC and COCO. For VQA, we adopt a fundamental architecture[6] proposed by [2] and a typical VQA dataset[7] which is established on the basis of COCO [9].

### 4.1. Validation for SHAP values

**Evaluation metrics.** Intuitively, an explanation image w.r.t. the target class $c$ can be generated using an original image $x$ and a related visual explanation map $L^c_{CAM}$ as below:

$$e^c = s(u(L^c_{CAM})) \circ x \tag{11}$$

where $u(\cdot)$ indicates the upsampling operation into the original input dimension and $s(\cdot)$ denotes a min-max normalization function. The operator $\circ$ refers to the Hadamard product. Hence, $e^c$ preserves the information of $x$ only in the region which $L^c_{CAM}$ considers important.

In general, $L^c_{CAM}$ is expected to recognize the regions which contribute the most to the model's decision. Thus, we can evaluate the faithfulness of $L^c_{CAM}$ on the object recognition task via the two metrics proposed in [4]: increase in confidence (IIC) and average drop (AD), which are defined as:

$$\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{[Y^c_i < O^c_i]} \times 100, \tag{12}$$

$$\frac{1}{N}\sum_{i=1}^{N} \frac{\max(0, Y^c_i - O^c_i)}{Y^c_i} \times 100, \tag{13}$$

respectively. $Y^c_i$ and $O^c_i$ are the model's softmax outputs of an $i$-th input image $x_i$ and the associated explanation image

[4] https://github.com/pytorch/vision/blob/master/torchvision
[5] https://github.com/facebookresearch/TorchRay
[6] https://github.com/tbmoon/basic_vqa
[7] https://visualqa.org/download.html

| Evaluation metric | Increase in confidence (%) | | | Average drop (%) | | | Average drop in deletion (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | ImageNet | VOC | COCO | ImageNet | VOC | COCO | ImageNet | VOC | COCO |
| SHAP-CAM$_1$ | 25.9 | 37.4 | 35.2 | 28.16 | 22.93 | 23.98 | 32.64 | 17.35 | 24.07 |
| SHAP-CAM$_{10}$ | 26.2 | 42.7 | 40.1 | 27.54 | 17.53 | 19.07 | 32.99 | 19.95 | 27.50 |
| SHAP-CAM$_{100}$ | **26.4** | **43.6** | **41.4** | **27.48** | **16.71** | **18.27** | **33.03** | **20.64** | **27.65** |

Table 1. Faithfulness evaluation on the object recognition task. We analyze 1,000 randomly selected images for each dataset. Higher is better for the increase in confidence and average drop in deletion. Lower is better for the average drop.

$e_i^c$, accordingly. $N$ denotes the number of images and $\mathbf{1}_{[\cdot]}$ is an indicator function. Higher is better for IIC and lower is better for AD.

However, IIC and AD evaluate the performance of the explanations via the *preservation* perspective; the region which is considered to be influential is maintained. We can also evaluate the performance using the opposite perspective (i.e., *deletion*); if we mute the region which is responsible for the target output, the softmax probability is expected to significantly drop. From this viewpoint, we suggest average drop in deletion (ADD) which can be defined as below:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{(Y_i^c - D_i^c)}{Y_i^c} \times 100 \tag{14}$$

where $D^c$ is the softmax output of the inverted explanation map $e_{inv}^c = (\mathbf{1} - s(u(L_{CAM}^c))) \circ x$. Higher is better for this metric.

**Faithfulness evaluation.** Table 1 presents the comparative results of IIC, AD, and ADD for SHAP-CAM$_1$, SHAP-CAM$_{10}$, and SHAP-CAM$_{100}$. We analyze 1,000 randomly selected images for each dataset. Furthermore, each case is averaged over 10 simulations for fair comparison. We can discover two important implications from Table 1. First, as the number of orderings increases, IIC and ADD increase and AD decreases, showing performance improvement. This result indicates that the closer the importance of the activation map is to the SHAP value, the more effectively the distinguishable region of the target object is found. Second, even compared to the other CAMs (see Table 3), SHAP-CAM$_{100}$ shows the best performances for all cases. It reveals the superiority of the SHAP attributions as the weight coefficients for CAM. However, this approach of averaging the marginal contributions of multiple orderings is impractical due to the significant computational burden. Therefore, we propose a cleverer approximation method: LIFT-CAM.

## 4.2. Approximation performance of LIFT-CAM

In this section, we quantitatively assess how precisely LIFT-CAM estimates the SHAP values of the activation maps. Since the exact SHAP values are unattainable, we regard the coefficients from SHAP-CAM$_{10k}$ as the true values for comparison (see the supplementary materials for the justification of this assumption). Table 2 shows the cosine similarities between the coefficients from state-of-the-art CAMs, including LIFT-CAM, and those from SHAP-CAM$_{10k}$. The similarity is averaged over 500 randomly selected images for each dataset.

| | ImageNet | VOC | COCO |
|---|---|---|---|
| Grad-CAM | 0.489 | 0.404 | 0.441 |
| Grad-CAM++ | 0.385 | 0.329 | 0.412 |
| Score-CAM | 0.195 | 0.181 | 0.157 |
| Ablation-CAM | 0.972 | 0.888 | 0.908 |
| XGrad-CAM | 0.969 | 0.865 | 0.877 |
| LIFT-CAM | **0.980** | **0.918** | **0.924** |

Table 2. The cosine similarities between the coefficients from CAMs and those from SHAP-CAM$_{10k}$. We randomly choose 500 images from each dataset for analysis.

As shown in Table 2, LIFT-CAM presents the highest similarities for all datasets (greater than 0.9), which indicates high relevance between the importance scores in LIFT-CAM and the SHAP values. Even if Ablation-CAM and XGrad-CAM also exhibit high similarities, they fall behind LIFT-CAM due to dissatisfaction of the local accuracy property. The other CAMs cannot approximate the SHAP values, providing consistently low similarities.

## 4.3. Performance evaluation of LIFT-CAM

The experimental results from Secs. 4.1 and 4.2 motivate us to evaluate the importances of activation maps with LIFT-CAM. To confirm the excellence of our LIFT-CAM, we compare the performances of the method with those of various state-of-the-art saliency methods.

### 4.3.1 Qualitative evaluation

Figure 3 provides qualitative comparisons between various saliency methods via visualization. Each row represents the visual explanation maps for each dataset. When compared to the other methods, our proposed method, LIFT-CAM, yields visually interpretable explanation maps for all cases. It clearly pinpoints the essential parts of the specific objects which are responsible for the classification results. This can be noted in the notebook case (row 1), for which most of the other visualizations cannot decipher the lower

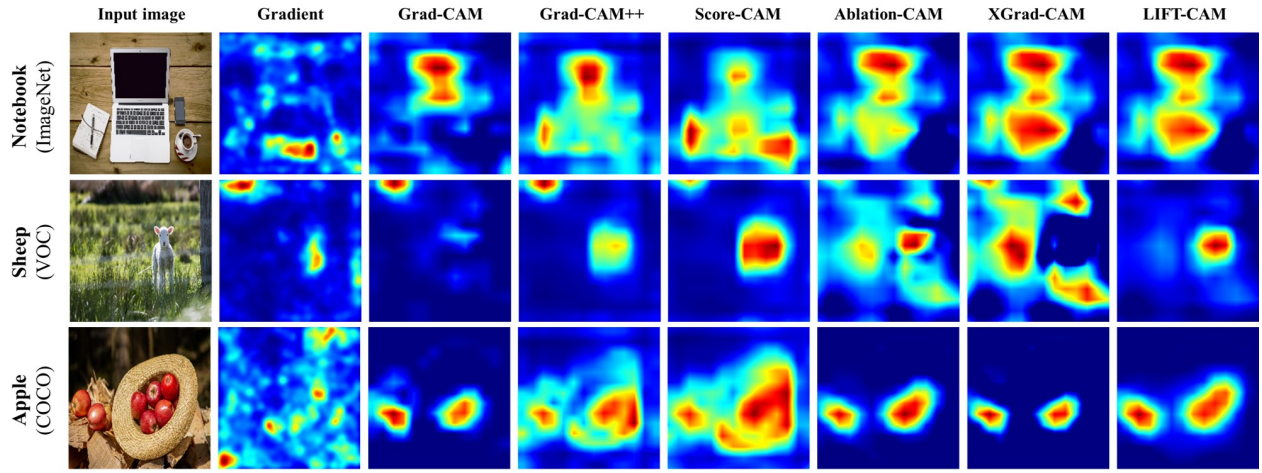


Figure 3. Visual explanation maps of state-of-the-art saliency methods and our proposed LIFT-CAM. Note that we apply a smoothing technique [7] to Gradient [18] to acquire plausible explanation maps.

| Evaluation metric | Increase in confidence (%) | | | Average drop (%) | | | Average drop in deletion (%) | | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | ImageNet | VOC | COCO | ImageNet | VOC | COCO | ImageNet | VOC | COCO |
| Grad-CAM | 24.0 | 32.7 | 31.9 | 31.89 | 30.73 | 30.74 | 30.60 | 17.43 | 25.66 |
| Grad-CAM++ | 23.1 | 33.8 | 33.5 | 30.53 | 17.20 | 20.87 | 27.98 | 15.85 | 24.16 |
| Score-CAM | 22.8 | 29.4 | 23.9 | 29.91 | 17.49 | 23.66 | 27.52 | 14.12 | 17.35 |
| Ablation-CAM | 24.1 | 34.4 | 35.0 | 29.41 | 25.49 | 23.99 | 32.52 | 19.42 | **26.75** |
| XGrad-CAM | 24.7 | 31.7 | 32.5 | 29.19 | 29.52 | 28.52 | 27.52 | 19.00 | 26.09 |
| LIFT-CAM | **25.2** | **38.7** | **39.3** | **29.15** | **17.15** | **18.65** | **32.95** | **20.09** | 26.34 |

Table 3. Comparative evaluation of faithfulness on the object recognition task among various CAMs. Results with 1,000 randomly sampled images are provided for each dataset. Higher is better for the increase in confidence and average drop in deletion. Lower is better for the average drop.

part of the notebook. Furthermore, LIFT-CAM alleviates pixel noise without highlighting trivial evidence. In the sheep case (row 2), the artifacts of the image are eliminated by LIFT-CAM and the exact location of the sheep is captured by the method. Last, the method successfully locates multiple objects in the apple case (row 3) by providing the object-focused map.

#### 4.3.2 Faithfulness evaluation

**IIC, AD, and ADD.** Table 3 shows the results of IIC, AD, and ADD of each CAM in various datasets. The three metrics can represent the object recognition performances of the saliency methods in a complementary way. LIFT-CAM provides the best results in most cases. Exceptionally, for ADD in COCO, Ablation-CAM presents a better result than LIFT-CAM. However, LIFT-CAM also provides a comparable result, exhibiting the negligible difference. In addition, it should be noted that LIFT-CAM is much faster than Ablation-CAM since it requires only a single backward pass to calculate the coefficients. In consequence, LIFT-CAM can accurately and efficiently determine which object is responsible for the model's prediction.

| | Insertion | Deletion |
|---|---|---|
| Grad-CAM | 0.4427 | 0.0891 |
| Grad-CAM++ | 0.4350 | 0.0969 |
| Score-CAM | 0.4345 | 0.1002 |
| Ablation-CAM | 0.4685 | 0.0873 |
| XGrad-CAM | 0.4680 | 0.0871 |
| LIFT-CAM | **0.4712** | **0.0866** |

Table 4. The AUC results in terms of insertion and deletion. The values are averaged over 1,000 randomly selected images from ImageNet. Higher is better for insertion and lower is better for deletion.

**Area under probability curve.** The above three metrics tend to be advantageous for methods which provide explanation maps of large magnitude. To exclude the influence of the magnitude, we can binarize the explanation map with two opposite perspectives: *insertion* and *deletion* [12]. We simply introduce or remove pixels from an image by setting

| | Grad-CAM | Grad-CAM++ | Score-CAM | Ablation-CAM | XGrad-CAM | LIFT-CAM |
|---|---|---|---|---|---|---|
| Proportion (%) | 47.76 | 49.14 | 51.14 | 51.87 | 51.85 | **52.43** |

Table 5. The proportions of energy located in bounding boxes for various CAMs. The values are averaged over 1,000 randomly selected images from ImageNet.

the pixel values of the explanation map to one or zero with step 0.025 (introduce or remove 2.5% pixels of the whole image in each step) and calculate the area under the probability curve (AUC). As shown in Table 4, LIFT-CAM provides the most reliable results, presenting the highest insertion AUC and the lowest deletion AUC. Through this experiment, we confirm that LIFT-CAM succeeds in sorting the pixels according to the contributions to the target result.

#### 4.3.3 Localization evaluation

It is reasonable to expect that a dependable explanation map overlaps with a target object. Therefore, we can also assess the reliability of the map via localization ability in addition to the softmax probability. [19] proposed a new localization metric, the energy-based pointing game, which is an improved version of the pointing game [20]. The metric gauges how much energy of the explanation map interacts with the bounding box of the target object. The metric can be formulated as follows:

$$\text{Proportion} = \frac{\sum_{(i,j)\in bbox} s(u(L^c_{CAM}))_{(i,j)}}{\sum_{(i,j)\in \Lambda'} s(u(L^c_{CAM}))_{(i,j)}} \tag{15}$$

where $\Lambda' = \{1, \ldots, H'\} \times \{1, \ldots, W'\}$ is an original image dimension. Therefore, $s(u(L^c_{CAM}))_{(i,j)}$ denotes a min-max normalized importance at pixel location $(i, j)$.

The proportions of the various methods are reported in Table 5. LIFT-CAM shows the highest proportion compared to the other methods. This implies that LIFT-CAM produces a compact explanation map which focuses on the essential parts of the images with less noise.

### 4.4. Application to VQA

We also apply LIFT-CAM to VQA to demonstrate the applicability of the method. We consider the standard VQA model [2] which consists of a CNN and a recurrent neural network in parallel. They function to embed images and questions, respectively. The two embedded vectors are fused and entered into a classifier to produce the answer.

Figure 4 illustrates the explanation maps of Grad-CAM and LIFT-CAM. As displayed in the figure, LIFT-CAM succeeds in highlighting the regions in the given images which are more relevant to the question and prediction pairs than Grad-CAM. Additionally, since this is a classification problem, the IIC, AD, and ADD of each method can be evaluated with fixed question embeddings. Table 6 represents the

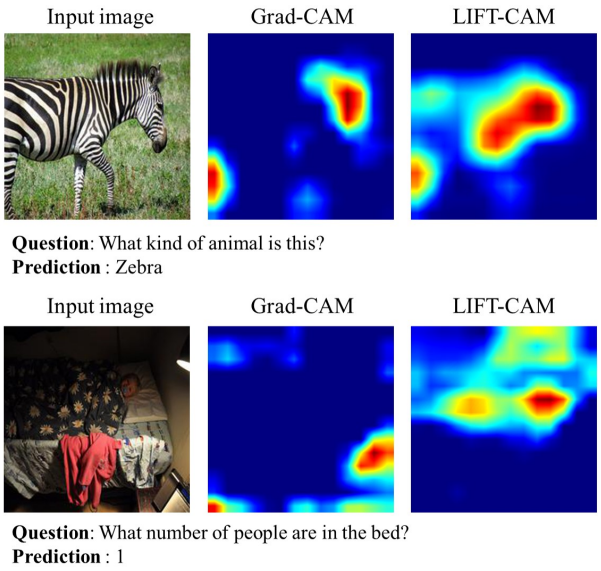


Figure 4. Visual explanation maps of Grad-CAM and LIFT-CAM for VQA.

| | Grad-CAM | LIFT-CAM |
|---|---|---|
| Increase in confidence (%) | 41.95 | **43.39** |
| Average drop (%) | 16.71 | **14.14** |
| Average drop in deletion (%) | 9.09 | **9.58** |

Table 6. Faithfulness evaluation for the VQA task. The values are computed through the complete validation sets (214,354 image and question pairs). Higher is better for the increase in confidence and average drop in deletion. Lower is better for the average drop.

comparison results between Grad-CAM and LIFT-CAM in terms of those metrics. As demonstrated in the table, LIFT-CAM provides performances superior to Grad-CAM for all of the metrics. This indicates that LIFT-CAM is better at figuring out the essential parts of images, which can serve as evidence for the answers to the questions.

## 5. Conclusion

In this work, we suggest a novel analytic framework to decide the weight coefficients of CAM using additive feature attribution methods. Based on the idea that SHAP values is a unified solution of the methods, we adopt the values as the coefficients and demonstrate their excellence. Moreover, we introduce LIFT-CAM, which approximates

the SHAP values of activation maps precisely with a single backward pass. It presents quantitatively and qualitatively enhanced visual explanations compared with the other previous CAMs.

## References


[1] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[3] 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.

[4] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.

[5] Saurabh Desai and Harish G Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 972–980. IEEE, 2020.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[7] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2950–2958, 2019.

[8] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. In *31th British Machine Vision Conference, BMVC 2020*.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[10] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.

[11] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[12] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *29th British Machine Vision Conference, BMVC 2018*.

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[14] 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.

[15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[16] Lloyd S Shapley. A value for n-person games. Technical report, Rand Corp Santa Monica CA, 1952.

[17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153, 2017.

[18] K. Simonyan, A. Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2014.

[19] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–25, 2020.

[20] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[21] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.