



Opti-CAM: Optimizing saliency maps for interpretability

Hanwei Zhang^{a,**}, Felipe Torres^a, Ronan Sicre^a, Yannis Avrithis^b, Stephane Ayache^a

^a*Centrale Marseille, Aix Marseille Univ, CNRS, LIS, Marseille, France*

^b*Institute of Advanced Research on Artificial Intelligence (IARAI)*

ABSTRACT

Methods based on *class activation maps* (CAM) provide a simple mechanism to interpret predictions of convolutional neural networks by using linear combinations of feature maps as saliency maps. By contrast, masking-based methods optimize a saliency map directly in the image space or learn it by training another network on additional data. In this work we introduce Opti-CAM, combining ideas from CAM-based and masking-based approaches. Our saliency map is a linear combination of feature maps, where weights are optimized per image such that the logit of the masked image for a given class is maximized. We also fix a fundamental flaw in two of the most common evaluation metrics of attribution methods. On several datasets, Opti-CAM largely outperforms other CAM-based approaches according to the most relevant classification metrics. We provide empirical evidence supporting that localization and classifier interpretability are not necessarily aligned.

© 2024 Elsevier Ltd. All rights reserved.

1. Introduction

The success of *deep neural networks* (DNN) and their increasing penetration into most sectors of human activity have led to a growing interest in understanding how these models make their predictions. Unlike shallow methods, DNN have a high complexity and it is not possible to directly explain their inference process in a human understandable manner. This challenge has opened up an entire research field [1, 2, 3, 4, 5].

In this work, we are interested in the interpretability of deep neural networks through the generation of *saliency maps*, highlighting regions of an image that are responsible for the prediction. This originates in *gradient-based* methods [6, 7], including variants of backpropagation [8, 9, 10]. CAM [11] introduced class-specific linear combinations of feature maps, and led to several alternative weighting schemes [12, 13, 14], including the use of gradients [15, 16]. On the other hand, *occlusion-* or *masking-based* methods [17, 18, 19, 20] remove regions in the image space while improving classification performance.

Score-CAM [13] is a CAM-based method where the weight of each feature map is defined by using that feature map as a mask and measuring the resulting increase of class score; hence,

it is both CAM-based and masking-based but does not use gradients. It resembles the numerical gradient approximation, in that it needs *one forward pass per weight*. Instead, the analytical approach would be to use a linear combination of feature maps as a mask, express the class score as a function of the weights and measure the gradient analytically, in a *single backward pass*. Then, *why not use gradient descent to maximize the class score?* Like Score-CAM, the optimal mask should highlight regions for which the network is most confident. But, as a result of the optimization process, this mask should be obtained in a more principled and efficient way.

Masking-based methods, such as extremal perturbations [19] or IBA [20], introduce a mask as a variable in the input or feature space, express the class score as a function of this mask and use gradient descent to maximize the class score. Because the variable being optimized is a high-dimensional image or tensor, additional constraints or regularizers are needed to control e.g. the smoothness and the salient area. This translates to more hyperparameters or more expensive optimization.

Motivated by the above, we introduce Opti-CAM, illustrated in Figure 1. We form a linear combination of feature maps, where the weights are a variable. Treating it as a saliency map, we form a masked version of the input image that is fed again to the network. Then, the logit of a given class for the masked version of the input is maximized to obtain the optimal weights. Thus, Opti-CAM can be seen as an analytical counterpart of

**Corresponding author

Score-CAM, where weights of the linear combination are updated by a single backward pass, thus can be optimized iteratively. It can also be seen as a masking-based method, where the mask to be optimized lies in the linear span of the feature maps, like CAM-based methods.

The evaluation metrics most relevant to using a saliency map as a mask are *average drop* (AD) and *average increase* (AI) [16]. The problem is that the two metrics are not defined in a symmetric way. As a result, there exists a trivial attribution method called Fake-CAM [21] that outperforms the state of the art in both metrics. To address this, we introduce the symmetric counterpart of AD, which we call *average gain* (AG), to be paired with AD as a replacement of AI. As expected, Fake-CAM fails AG.

In summary, we make the following contributions:

1. We introduce Opti-CAM, a simple model for saliency map generation that combines ideas from CAM-based and masking-based approaches. Opti-CAM does not need any extra data, network or training.
2. Compared with gradient-free methods [13, 22, 12], it finds the optimal feature map weights and is on par or faster, assuming that the number of iterations is less than the number of channels.
3. We introduce a new evaluation metric, *average gain* (AG), to be paired with *average drop* (AD) as a replacement of *average increase* (AI) [16].
4. On several datasets, we improve the state of the art by a large margin, reaching near-perfect performance according to the most relevant classification metrics.
5. We shed more light into how a classifier may exploit background context.

2. Related Work

A large number of works study *explainability*, *interpretability* or *attribution* of machine learning models, especially DNN [1, 2, 3, 4, 5]. These works can be categorized into *transparency* and *post-hoc interpretability* [23, 1]. The former addresses how to design an internally understandable model. Here we are interested in the latter, which keeps the studied network fixed and interprets its inner processing [24, 25, 26]. Among post-hoc methods, LIME [24] and SHAP [25] are well-known model-agnostic methods that rate feature importance. More specifically, we are interested in the generation of *saliency maps*. These methods are mostly based on gradients, CAM [11], occlusion, or a combination.

Gradient-based methods. Gradient-based methods [27, 9, 28] use the gradient of a target class score with respect to the input to measure the effect of different image regions on the prediction. In [6], the gradient is directly treated as a saliency map. Inspired by DeconvNet [8], *guided backpropagation* [9] improves the explanation by setting negative gradients to zero using ReLU units. Other methods [29, 30, 31] are inspired by Layer-wise Relevance Propagation (LRP) [10]. SmoothGrad [32] and *integrated gradients* [33] accumulate gradients

into saliency maps, while NormGrad [34] attempts to unify gradient-based methods.

A different approach is to use *adversarial attacks* [35, 36]. These search along the gradient to identify pixels that, when modified, lead to changes in the model prediction. The identified pixels are considered significant for the prediction. Gradient-based methods do not always satisfy the fundamental property of implementation invariance; for example, DeepLift and LRP are not always identical for two functionally equivalent networks [33].

CAM-based methods. *Class activation maps* (CAM) [11] is a visualization method that highlights the image regions most relevant to a target class by a linear combination of feature maps. A number of variants use different definitions of weights. Many rely on gradients, including GradCAM [15], GradCAM++ [16], and XGradCAM [37]. Gradient-free methods, including Ablation-CAM [12] and Score-CAM [13], rather measure the effect on the target class score of each feature map acting as a mask on the input. Additional methods can be found in the survey conducted by He et al. (2022) [38]. We inherit the idea of masking but for linear combinations of feature maps and we iteratively optimize the coefficients by analytical gradient computation. Our method is thus faster when the number of iterations is less than the number of channels.

Occlusion (masking)-based methods. These methods use a number of candidate masks, measure their effect on the prediction, then combine them in a single saliency map. LIME [24] approximates the behavior of complex models through the creation of local surrogate models based on perturbations of inputs. LIME relies on superpixels to define the fundamental elements for perturbation and masking. This approach proves either too computationally intensive or too coarse for high-resolution images. Based on game theory, SHAP [25] provides a unified framework to estimate the importance of inputs through SHAP values. SHAP-based methods also face the challenge of balancing performance with computational complexity.

RISE [22] randomly masks input images and uses the class score as a weight to define a linear combination. Some occlusion-based methods are combined with gradients. *Meaningful perturbations* [18] and *extremal perturbations* [19] directly optimize the mask in the image space by using gradients. They require a large number of parameters as well as regularizers, e.g. for smoothness. *Information bottleneck attribution* (IBA) [20] optimizes the mask in the feature space as a tensor instead. Score-CAM [13] is also an occlusion-based method, using individual feature maps as candidate masks. The same holds for our Opti-CAM, but for candidate masks constrained in the linear span of the feature maps. Compared with [19, 20], we have fewer parameters and do not require a regularizer.

Learning-based methods. While occlusion-based methods compute or optimize a mask for a particular image at inference, learning-based methods use an additional network or branch and they train it on extra data and image-level labels to predict a saliency map given an input image. This includes for example generators [39] or auto-encoders [17, 40, 41]. This

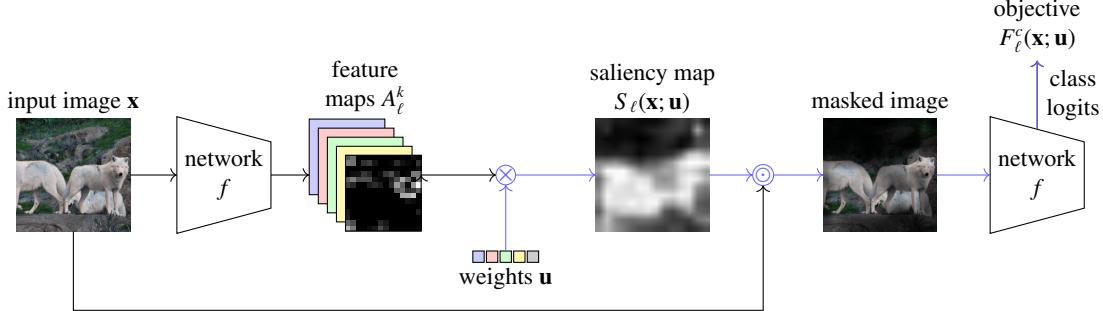


Fig. 1. Overview of Opti-CAM. We are given an input image \mathbf{x} , a fixed network f , a target layer ℓ and a class of interest c . We extract the feature maps from layer ℓ and obtain a saliency map $S_\ell(\mathbf{x}; \mathbf{u})$ by forming a convex combination of the feature maps (\times) with weights determined by a variable vector \mathbf{u} (8). After upsampling and normalizing, we element-wise multiply (\odot) the saliency map with the input image to form a “masked” version of the input, which is fed to f . The objective function $F_\ell^c(\mathbf{x}; \mathbf{u})$ measures the logit of class c for the masked image (10). We find the value of \mathbf{u}^* that maximizes this logit by optimizing along the path highlighted in blue (9), as well as the corresponding optimal saliency map $S_\ell(\mathbf{x}; \mathbf{u}^*)$ (11).

approach may be compared with weakly-supervised object detection [42], segmentation [43] or instance segmentation [44]. IBA [20] includes a learning-based approach in the feature space. Apart from requiring extra data, it is not satisfying in the sense that the learned decoder would need to be explained too. Our method does not need any extra data, network, or training.

Evaluation of attribution methods. Evaluating saliency maps is challenging because no ground truth attributions exist. Some studies utilize object segment masks as attribution ground truth, as noted by [45], yet this approach primarily concerns localization. Other studies use gaze density fixation maps as a reference for ground truth [45]. We contend that these ground truth methods solely provide information about images. While comparing saliency maps to these references aids in understanding the degree of alignment between network cognition and human perception, it overlooks crucial network information. We advocate that the ground truth should incorporate both image and network information. Such ground truth does not exist.

In the evaluation of attribution methods, metrics are categorized according to several key properties, including faithfulness, robustness, localization, complexity and randomization [46]. Each property carries distinct assumptions regarding the saliency maps generated by these methods. Faithfulness, in particular, stands out as it directly correlates with the accuracy of the classification score. Given its direct relationship to model performance, we prioritize this category as it provides crucial insights into how faithfully the attribution method represents the model’s decision-making process. By focusing on faithfulness, we aim to ensure that the attribution method accurately captures the reasoning of the model.

Average drop (AD) and *average increase* (AI), also known as increase in confidence [16] are well-established metrics. They consider the effect on the predicted class probabilities by masking the input image with the saliency map. There is a fundamental flaw in using AD, AI as a pair of metrics, which we fix by replacing AI by a new metric, *average gain* (AG).

Insertion (I) and *deletion* (D) sequentially insert or delete pixels by decreasing order of saliency and observe the effect on the prediction. The resulting images are out-of-distribution (OOD) [47] and the metrics favor small and compact regions. Localization metrics measure how the saliency maps

are aligned with object bounding boxes, which ignores the importance of background context [48, 49]. We demonstrate that localization and attribution are not well-aligned as tasks.

3. Opti-CAM

3.1. Preliminaries

Notation. Consider a classifier network $f : \mathcal{X} \rightarrow \mathbb{R}^C$ that maps an input image $\mathbf{x} \in \mathcal{X}$ to a logit vector $\mathbf{y} = f(\mathbf{x}) \in \mathbb{R}^C$, where \mathcal{X} is the image space and C is the number of classes. We denote by $y_c = f(\mathbf{x})_c$ the predicted logit and by $p_c = \text{softmax}(\mathbf{y})_c := e^{y_c} / \sum_j e^{y_j}$ the predicted probability for class c . For layer ℓ with K_ℓ channels, we denote by $A_\ell^k = f_\ell^k(\mathbf{x}) \in \mathbb{R}^{h_\ell \times w_\ell}$ the feature map for channel $k \in \{1, \dots, K_\ell\}$, with spatial resolution $h_\ell \times w_\ell$. Because of relu non-linearities, we assume that feature maps are non-negative. Similarly, we denote by $S_\ell \in \mathbb{R}^{h_\ell \times w_\ell}$ a 2D saliency map.

Background: CAM-based saliency maps. Given a layer ℓ and a class of interest c , we consider saliency maps given by the general formula

$$S_\ell^c(\mathbf{x}) := h \left(\sum_k w_k^c A_\ell^k \right), \quad (1)$$

where w_k^c are weights defining a linear combination over channels and h is an activation function. CAM [11] is defined for the last layer L only with h being the identity mapping and w_k^c being the classifier weight connecting the k -th channel with class c . Grad-CAM [15] is defined for any layer ℓ with $h = \text{relu}$ and weights

$$w_k^c := \text{GAP} \left(\frac{\partial y_c}{\partial A_\ell^k} \right), \quad (2)$$

where GAP is global average pooling. The motivation for relu is that we are only interested in features that have a positive effect on the class of interest, *i.e.* pixels whose intensity should be increased in order to increase y_c .

Score-CAM [13] is also defined for any layer ℓ with $h = \text{relu}$ and weights $w_k^c := \text{softmax}(\mathbf{u}^c)_k$. Softmax normalization considers positive channel contributions only and attends to few feature maps. Here, vector $\mathbf{u}^c \in \mathbb{R}^{K_\ell}$ measures the increase in

confidence for class c that compares a known baseline image \mathbf{x}_b with the input image \mathbf{x} masked according to feature map A_ℓ^k , for all channels k :

$$u_k^c := f\left(\mathbf{x} \odot n\left(\text{up}(A_\ell^k)\right)\right)_c - f(\mathbf{x}_b)_c, \quad (3)$$

where \odot is the Hadamard product. For this to work, the feature map A_ℓ^k is adapted to \mathbf{x} first: up denotes upsampling to the spatial resolution of \mathbf{x} and

$$n(A) := \frac{A - \min A}{\max A - \min A} \quad (4)$$

is a normalization of matrix A into $[0, 1]$. While Score-CAM does not need gradients, it requires as many forward passes through the network as the number of channels in the chosen layer, which is computationally expensive.

Motivation. Score-CAM considers each feature map as a mask in isolation. How about linear combinations? Given a vector $\mathbf{w} \in \mathbb{R}^{K_\ell}$ with w_k its k -th element, let

$$F(\mathbf{w}) := f\left(\mathbf{x} \odot n\left(\text{up}\left(\sum_k w_k A_\ell^k\right)\right)\right)_c. \quad (5)$$

If we assume that $\mathbf{x}_b = \mathbf{0}$ in (3) and define $n(\mathbf{0}) := \mathbf{0}$ in (4), then we can rewrite the right-hand side of (3) as

$$\frac{F(\mathbf{w}_0 + \delta \mathbf{e}_k) - F(\mathbf{w}_0)}{\delta}, \quad (6)$$

where $\mathbf{w}_0 = \mathbf{0}$, $\delta = 1$ and \mathbf{e}_k is the k -th standard basis vector of \mathbb{R}^{K_ℓ} . This resembles the numerical approximation of the derivative $\frac{\partial F}{\partial w_k}(\mathbf{w}_0)$, except that δ is not small as usual. One could compute derivatives efficiently by standard backpropagation instead. It is then possible to iteratively optimize F with respect to \mathbf{w} , starting at any \mathbf{w}_0 .

As an alternative, consider masking-based methods relying on optimization in the input space, like *meaningful perturbations* (MP) [18] or *extremal perturbations* [19]. In general, optimization takes the form

$$S^c(\mathbf{x}) := \arg \max_{\mathbf{m} \in \mathcal{M}} f(\mathbf{x} \odot n(\text{up}(\mathbf{m})))_c + \lambda R(\mathbf{m}). \quad (7)$$

Here, a mask \mathbf{m} is directly optimized and does not rely on feature maps, hence the saliency map $S^c(\mathbf{x})$ is not connected to any layer ℓ . The mask is at the same or lower resolution than the input image. In the latter case, upsampling is still necessary.

In this approach, one indeed computes derivatives by backpropagation and indeed iteratively optimizes \mathbf{m} . However, because \mathbf{m} is high-dimensional, there are constraints expressed by $\mathbf{m} \in \mathcal{M}$, e.g. \mathbf{m} has a certain norm, and regularizers like $R(\mathbf{m})$, e.g. \mathbf{m} is smooth in a certain way. This makes optimization harder or more expensive and introduces more hyperparameters like λ . One could simply constrain \mathbf{m} to lie in the linear span of $\{A_\ell^k\}_{k=1}^{K_\ell}$ instead, like all CAM-based methods.

3.2. Method

Saliency maps. As motivated by section 3.1, we obtain a saliency map as a convex combination of feature maps by optimizing a given objective function with respect to the weights.

In particular, following [13], we use channel weights $w_k := \text{softmax}(\mathbf{u})_k$, where $\mathbf{u} \in \mathbb{R}^{K_\ell}$ is a variable. We then consider saliency map S_ℓ in layer ℓ as a function of both the input image \mathbf{x} and variable \mathbf{u} :

$$S_\ell(\mathbf{x}; \mathbf{u}) := h\left(\sum_k \text{softmax}(\mathbf{u})_k A_\ell^k\right). \quad (8)$$

Comparing with (1), h can be any function including the identity function or the activation function used by the network, because we normalize S_ℓ to make it non-negative.

Optimization. Now, given a layer ℓ and a class of interest c , we find the vector \mathbf{u}^* that maximizes the classifier confidence for class c , when the input image \mathbf{x} is masked according to saliency map $S_\ell(\mathbf{x}; \mathbf{u}^*)$:

$$\mathbf{u}^* := \arg \max_{\mathbf{u}} F_\ell^c(\mathbf{x}; \mathbf{u}), \quad (9)$$

where we define the objective function

$$F_\ell^c(\mathbf{x}; \mathbf{u}) := g_c(f(\mathbf{x} \odot n(\text{up}(S_\ell(\mathbf{x}; \mathbf{u}))))). \quad (10)$$

Here, the saliency map $S_\ell(\mathbf{x}; \mathbf{u})$ is adapted to \mathbf{x} exactly as in (3) in terms of resolution and normalization. For *normalization function* n , the default is (4). The *selector function* g_c operates on the logit vector \mathbf{y} ; the default is to select the logit of class c , i.e. $g_c(\mathbf{y}) := y_c$. Other choices, including the definition of F_ℓ^c itself, are investigated in subsection 5.5 and in the supplementary material.

Opti-CAM. Putting everything together, we define

$$S_\ell^c(\mathbf{x}) := S_\ell(\mathbf{x}; \mathbf{u}^*) = S_\ell\left(\mathbf{x}; \arg \max_{\mathbf{u}} F_\ell^c(\mathbf{x}; \mathbf{u})\right), \quad (11)$$

where S_ℓ and F_ℓ^c are defined by (8) and (10) respectively. The objective function F_ℓ^c (10) depends on variable \mathbf{u} through S_ℓ (8), where the feature maps $A_\ell^k = f_\ell^k(\mathbf{x})$ are fixed. Then, (10) involves masking and a forward pass through the network f , which is also fixed.

Figure 1 is an abstract illustration of our method, called Opti-CAM, without details like upsampling and normalization (10). Optimization takes place along the highlighted path from variable \mathbf{u} to objective function F_ℓ^c . The saliency map is real-valued and the entire objective function is differentiable in \mathbf{u} . We use Adam optimizer [50] to solve the optimization problem (9).

Discussion. By maximizing (10), the saliency map focuses on the regions contributing to class c , while masked regions contribute less. This way, the influence of background in the average pooling process is reduced.

The saliency map is expressed as a linear combination of feature maps (8), with normalized weights. Hence, the saliency map is discouraged from taking up the entire image, both by the softmax competition (8) and by the fact that feature maps only respond to particular locations.

In case $g_c(\mathbf{y}) := y_c$, (11) takes the form of direct masking (7) with $R(\mathbf{m}) = \mathbf{0}$ and

$$\mathcal{M} := \{S_\ell(\mathbf{x}; \mathbf{u}) : \mathbf{u} \in \mathbb{R}^{K_\ell}\}. \quad (12)$$

This constraint makes ours a CAM-based method. It dispenses the need for regularizers, because we only optimize one vector over the feature dimensions (up to 2,048 for ResNet50), which is small compared with the dimensions of input images (50k for ImageNet). In addition, it does not complicate the optimization process in any way. It is only a different parametrization.

4. Average Gain (AG)

Average drop (AD) and average increase (AI) [16] are well-established classification metrics. They measure the effect on the predicted class probabilities by masking the input image with the saliency map. Let p_i^c and o_i^c be the predicted probability for class c given as input the i -th test image \mathbf{x}_i and its masked version respectively. Masking refers to element-wise multiplication with the saliency map, which is at the same resolution as the original image with values in $[0, 1]$. Let N be the number of test images. Class c is taken as the ground truth.

Average drop (AD) quantifies how much predictive power, measured as class probability, is lost when we only mask the image; lower is better:

$$\text{AD}(\%) := \frac{1}{N} \sum_{i=1}^N \frac{[p_i^c - o_i^c]_+}{p_i^c} \cdot 100. \quad (13)$$

Average increase (AI), also known as *increase in confidence*, measures the percentage of images where the masked image yields a higher class probability than the original; higher is better:

$$\text{AI}(\%) := \frac{1}{N} \sum_i^N \mathbb{1}_{p_i^c < o_i^c} \cdot 100. \quad (14)$$

AD and AI are not defined in a symmetric way. AD measures changes in class probability whereas AI measures a percentage of images. It is possible that the percentage is high while the actual increase is small. Hence, it is possible that an attribution method improves both. Indeed, [21] observes that a trivial method called Fake-CAM outperforms state-of-the-art methods, including Score-CAM, by a large margin. Fake-CAM simply defines a saliency map where the top-left pixel is set to zero and is uniform elsewhere. This questions the purpose of AD and AI.

Although the authors of [21] make this impressive observation, they use it to motivate the definition of a number of metrics that are orthogonal to the task at hand, *i.e.* measuring the effect of masking to the classifier. By contrast, we address the problem by introducing a new metric to be paired with AD as a replacement of AI. We define the new metric as follows.

Average gain (AG) quantifies how much predictive power, measured as class probability, is gained when we mask the image; higher is better:

$$\text{AG}(\%) := \frac{1}{N} \sum_{i=1}^N \frac{[o_i^c - p_i^c]_+}{1 - p_i^c} \cdot 100. \quad (15)$$

This definition is symmetric to the definition of average drop, in the sense that in absolute value, the numerator in the sum of

AD, AG is the positive and negative part of $p_i^c - o_i^c$ respectively and the denominator is the maximum value that the numerator can get as a function of o_i^c , given that $0 < o_i^c < p_i^c$ and $p_i^c < o_i^c < 1$ respectively. The two metrics thus compete each other, in the sense that changing o_i^c to improve one leaves the other unchanged or harms it. As we shall see, an extreme example is Fake-CAM, which yields near-perfect AD but fails completely on AG.

5. Experiments

We evaluate Opti-CAM and compare it quantitatively and qualitatively against other state-of-the-art methods on a number of datasets and networks. We report classification metrics with execution times and we provide visualizations, an ablation study and a study on the suitability of localization ground truth. A sanity check, additional classification results, localization metrics, more ablations, more visualizations and code are given in supplementary material.

5.1. Datasets and Networks

ImageNet. We use the validation set of ImageNet ILSVRC 2012 [51, 52], which contains 50,000 images evenly distributed over the 1,000 categories. For the ablation study and for timing, we sample 1,000 images from this set. Concerning the localization experiments, bounding boxes from the localization task of ILSVRC¹ are used on the same validation set.

Medical data. We use two medical image datasets, namely *Chest X-ray* [53] and *Kvasir* [54]. Complete qualitative and quantitative results are given in the supplementary. Here we only provide visualizations.

Networks. For all datasets, we use the pretrained ResNet50 [55] and VGG16 [56] networks with batch normalization [57] from the Pytorch model zoo². For ImageNet, we further use the pretrained ViT-B (16-224) [58] and DeiT-B (16-224) [59] from Pytorch image models (timm)³. We set h as the identify mapping. For ResNet50 and VGG16, all the values of $S_\ell(\mathbf{x}; \mathbf{u})$ are non-negative due to ReLU. For ViT-B and DeiT-B, there are negative values. In either case, S_ℓ is subsequently normalized to $[0, 1]$. On medical datasets, we fine-tune the networks as discussed in the supplementary material, where we also provide the setting details.

5.2. Evaluation

Metrics. We use *average drop* (AD) and *average increase* (AI) [16] metrics, as well as the proposed *average gain* (AG), to measure the effect on classification performance of masking the input image by a saliency map. In the supplementary, we also report *insertion* (I) and *deletion* (D) [22] and highlight their limitations. Using classification metrics, we show the limitations

¹<https://www.image-net.org/challenges/LSVRC/2012/index.php>

²<https://pytorch.org/vision/0.8/models.html>

³<https://github.com/rwightman/pytorch-image-models>

METHOD	RESNET50				VGG16			
	AD↓	AG↑	AI↑	T	AD↓	AG↑	AI↑	T
Fake-CAM [21]	0.8	1.6	46.0	0.00	0.5	0.6	42.6	0.00
Grad-CAM [15]	12.2	17.6	44.4	0.03	14.2	14.7	40.6	0.02
Grad-CAM++ [16]	12.9	16.0	42.1	0.03	17.1	10.2	33.4	0.02
Score-CAM [13]	8.6	26.6	56.7	15.22	13.5	15.6	41.7	3.11
Ablation-CAM [12]	12.5	16.4	42.8	18.26	15.5	12.6	36.9	2.98
XGrad-CAM [37]	12.2	17.6	44.4	0.03	13.8	14.8	41.2	0.02
Layer-CAM [60]	15.6	15.0	38.8	0.08	48.9	3.1	13.5	0.07
ExPerturbation [19]	38.1	9.5	22.5	152.96	43.0	7.1	20.5	83.20
HiRes-CAM [61]	12.2	17.6	44.4	0.03	15.8	13.2	37.8	0.02
LIME [24]	66.7	3.1	10.2	0.63	62.7	2.9	11.1	0.67
GradientShap [25]	96.8	0.1	1.3	0.87	94.0	0.1	2.9	0.60
Opti-CAM (ours)	1.5	68.8	92.8	4.15	1.3	71.2	92.7	3.94

Table 1. Classification metrics on ImageNet validation set, using CNNs. AD/AI: average drop/increase [16]; AG: average gain (ours); ↓ / ↑: lower / higher is better; T: Average time (sec) per batch of 8 images. Bold: best, excluding Fake-CAM.

of using the localization ground truth for the evaluation of attribution methods. In the supplementary, we provide a number of localization metrics from the *weakly-supervised object localization* (WSOL) task of ILSVRC2014⁴.

Methods. We compare against the following state-of-the-art methods: Grad-CAM [15], Grad-CAM++ [16], Score-CAM [13], Ablation-CAM [12], XGrad-CAM [37], Layer-CAM [60], ExtremalPerturbation [19], HiRes-CAM [61], LIME [24] and GradientSHAP [25]. Implementations are obtained from the PyTorch CAM library⁵ or TorchRay⁶. We use the official code of LIME⁷ and the package Captum⁸. For transformer models, we also compare against raw attention [58], rollout [62] and TIBAV [63]⁹.

Image normalization. It is standard that images are normalized before feeding them to a network. By doing so however, we cannot reproduce the results published for the baseline methods; rather, all results are improved dramatically. We can obtain results similar to published ones by *not* normalizing. We believe normalization is important and we include it in all our experiments. In the supplementary, we provide more details and results without normalization, as well as code that allows for reproduction and verification of our results.

5.3. Image classification

Opti-CAM is evaluated quantitatively using classification metrics and qualitatively by visualizing saliency maps.

5.3.1. AD/AI/AG

CNN. Table 1 shows ImageNet classification metrics using VGG16 and RESNET50. Our Opti-CAM brings impressive

METHOD	ViT-B				DeiT-B			
	AD↓	AG↑	AI↑	T	AD↓	AG↑	AI↑	T
Fake-CAM [21]	0.3	0.4	48.3	0.00	0.6	0.3	44.6	0.00
Grad-CAM [15]	69.4	2.5	12.4	0.14	33.5	1.7	12.5	0.11
Grad-CAM++ [16]	86.3	1.5	1.0	0.15	50.7	0.9	7.2	0.13
Score-CAM [13]	32.0	6.2	33.0	23.69	53.6	2.2	12.2	22.47
XGrad-CAM [37]	88.1	0.4	4.3	0.13	80.5	0.3	4.1	0.12
Layer-CAM [60]	82.0	0.2	2.9	0.24	88.9	0.4	2.6	0.24
ExPerturbation [19]	28.8	6.2	24.4	133.52	60.9	2.0	8.5	129.12
RawAtt [58]	92.6	0.2	2.8	0.02	95.3	0.0	1.8	0.02
Rollout [62]	42.1	5.6	20.9	0.02	55.2	0.8	7.9	0.02
TIBAV [63]	81.7	0.8	5.8	0.16	62.3	0.7	7.1	0.16
HiRes-CAM [61]	98.4	0.0	0.7	0.03	97.2	0.0	1.2	0.03
LIME [24]	71.5	1.7	7.1	0.43	55.6	1.7	9.7	0.42
GradientShap [25]	98.1	0.0	0.8	1.36	95.8	0.1	1.5	0.71
Opti-CAM (ours)	0.6	18.0	90.1	16.05	0.9	26.0	83.5	15.17

Table 2. Classification metrics on ImageNet validation set, using transformers. AD/AI: average drop/increase [16]; AG: average gain (ours); ↓ / ↑: lower / higher is better. T: Average time (sec) per batch of 8 images. Bold: best, excluding Fake-CAM.

performance in terms of average drop (AD) and Average Increase (AI) metrics. That is, not only impressive improvement over baselines, but near-perfect: near-zero AD and above 90% AI. Our new metric AG is lower, around 70% for Opti-CAM, but this is still several times higher than for all the other methods.

Interestingly, Fake-CAM [21] is the winner in terms of AD and second or third best in AI after Opti-CAM and Score-CAM, but fails completely AG. This is expected and makes Fake-CAM uninteresting as it should be: By only masking one pixel, the classification score can hardly drop (0.8% on ResNet50) and while it increases very often (on 46% of images), the gain is as little as the drop (0.7%). This makes the pair (AD, AG) sufficient as primary metrics and AI can be thought of as secondary, if important at all.

Due to computation limits, LIME fails because the superpixels are too coarse compared to other methods. We choose GradientSHAP as a representative of SHAP methods, which is more suitable for high-resolution images. Nonetheless, GradientSHAP treats individual pixels as the basic unit, which does not yield satisfactory results for classification metrics, namely AI/AG/AD/I/D.

Table 1 also includes average execution time per image over the 1000-image ImageNet subset for all methods. Opti-CAM is slower than gradient-based methods that require only one pass through the network, but on par or faster than gradient-free methods. Indeed, we use a maximum of 100 iterations with one forward/backward pass per iteration, while Score-CAM and Ablation-CAM perform as many forward passes as channels. Hence they are much slower on ResNet50 than VGG16. ExtremalPerturbation does not depend on the number of channels but is very slow by performing a complex optimization in the image space.

Transformers. Table 2 shows ImageNet classification metrics using ViT and DeiT. Unlike CAM-based methods that rely on a class-specific linear combination of feature maps, raw attention [58] and rollout [62] use the attention map of the [CLS] token from the last attention block and from all blocks respectively. This attention map depends only on the particular image

⁴<https://www.image-net.org/challenges/LSVRC/2014/index#>

⁵<https://github.com/jacobgil/pytorch-grad-cam>

⁶<https://github.com/facebookresearch/TorchRay>

⁷<https://github.com/marcotcr/lime/tree/master>

⁸<https://github.com/pytorch/captum/tree/master>

⁹<https://github.com/hila-chefer/Transformer-Explainability>

and not on the target class, hence it is not really comparable. TIBAV [63] uses both instance-specific and class-specific information.

Opti-CAM outperforms all other methods dramatically, reaching near-zero AD and AI above 80 or 90%. According to our new AG metric, Opti-CAM still works while all other methods fail, but AG is much more conservative than AI. On ViT-B for example, the classification score increases for 90.1% of the images by masking with Opti-CAM, but the gain is only 18.0% on average.

Visualization. Figure 2 illustrates saliency map examples from ImageNet, Chest X-ray and Kvasir datasets. Opti-CAM saliency map is in general more spread out. This better highlights full objects, multiple instances or background context, which may be taken into account by the model. On Chest X-ray, Opti-CAM and Score-CAM are the only methods that capture the chest, while all others focus on image corners. More examples on datasets and networks as well as quantitative evaluation on medical data are given in the supplementary material.

5.3.2. Insertion/Deletion

Definition. Insertion/Deletion [22] are based on the probability $p_i^{c_p}$ for the predicted class c_p as pixels are “inserted” or “deleted” from image \mathbf{x}_i , averaged over the number of pixels and over all images in the test set.

Deletion measures the decrease in the probability of class c_p when removing pixels one by one in decreasing order of saliency, where removal is taken as setting the value to zero; lower is better.

Insertion, by contrast, measures the increase in the probability of class c_p when adding pixels, again by decreasing order of saliency. In this case, we begin with a version of the image that is distorted by Gaussian blur and then addition is taken as setting the value of the pixel according to the original image. Higher is better.

Results. The experimental results are shown in Table 3 for CNNs and transformers. ExPerturbation [19] is expected to perform best in insertion because its optimization objective is very similar to this evaluation metric, using blurring for masked regions. However, ExPerturbation [19] only performs best on ResNet50. TIBAV [63], which is designed for transformers, outperforms the other methods on DeiT and ViT. According to the results of Insertion/Deletion, Opti-CAM has low performance but there is no clear winner on either CNNs or transformers.

To further understand the behavior of Opti-CAM, we investigate in Figure 3 examples where Score-CAM succeeds (insertion score greater than 90 and deletion score less than 10) and Opti-CAM fails (insertion score less than 70 and deletion score greater than 15). Compared with Score-CAM, the saliency maps obtained by Opti-CAM are more spread out and highlight several parts of the object and background context. In most of the cases, Opti-CAM fails I/D because it not only finds the object but also attaches importance to the background.

We argue that this is not a failure. As our localization experiment in Table 5 indicates, the background is useful in discrim-

METHOD	RESNET50		VGG16		ViT-B		DeiT-B	
	I↑	D↓	I↑	D↓	I↑	D↓	I↑	D↓
Fake-CAM [21]	50.7	28.1	46.1	26.9	57.4	33.3	57.5	34.2
Grad-CAM [15]	66.3	14.7	64.1	11.6	62.9	19.8	61.8	17.5
Grad-CAM++ [16]	66.0	14.7	62.9	12.2	56.7	29.3	60.5	21.9
Score-CAM [13]	65.7	16.3	62.5	12.1	66.5	15.1	60.6	24.4
XGrad-CAM [37]	66.3	14.7	64.1	11.7	55.6	26.5	55.2	31.1
Layer-CAM [60]	67.0	14.2	58.3	6.4	62.9	14.6	61.6	21.2
ExPerturbation [19]	70.7	15.0	61.1	15.0	64.4	18.4	62.1	27.0
Ablation-CAM [12]	65.9	14.6	63.8	11.4	-	-	-	-
RawAtt [58]	-	-	-	-	62.2	17.9	56.3	29.3
Rollout [62]	-	-	-	-	64.8	15.2	56.7	32.8
TIBAV [63]	-	-	-	-	66.1	14.1	63.7	16.3
LIME [24]	60.0	15.3	56.6	14.8	61.4	22.9	62.1	27.8
GradientShap [25]	58.5	6.3	43.4	5.2	59.2	16.6	57.6	22.3
Opti-CAM (ours)	62.0	19.7	59.2	11.0	60.5	22.0	59.2	22.8

Table 3. I/D: insertion/deletion [22] scores on ImageNet validation set; ↓ / ↑: lower / higher is better.

inating a class. Often, the network recognizes the background better than the object itself. For example, a gas pump is likely to be seen with a truck, and a hare is often seen on grass. Several parts of the object are highlighted by Opti-CAM for the worm fence, terrier dog, hare, and manhole cover. Finally, several instances of spaniel dog are found by Opti-CAM.

Insertion/Deletion include 224 steps of binarization, with a set of 224 pixels being inserted/deleted at each step. If these pixels are all inserted over a single small area, the effect on the classifier is more immediate than when sparsely inserting pixels over multiple areas. The same observation holds for deletion. By contrast, Opti-CAM attempts to find regions that contribute to the classification as a whole. There is no guarantee that those regions are effective when used in isolation.

5.3.3. More metrics

In this section, we show additional metrics including AOPC [64], Max-Sensitivity [65] and ADCC [21].

We use the code and suggested parameters of package Quantus¹⁰ to measure AOPC and MS. In particular, patch size 14 and number of evaluation regions 10 for AOPC; lower bound 0.2 and number of samples 10 for MS. For ADCC, we use the official code¹¹. We evaluate these metrics on ImageNet validation set using ResNet50 and VGG16. The results are reported in Table 4. Since AOPC shares the same philosophy as I/D, it is not a surprise that Opti-CAM has poor performance on AOPC. Opti-CAM achieves the best performance on MS.

5.4. Object localization

Localization metrics are used to measure the precision of saliency maps relative to ground truth bounding boxes of the foreground object of interest. These metrics originate from weakly supervised localization (WSOL). However, the objectives of WSOL and explaining the decision of a DNN are not necessarily aligned, since context may play an important role in the decision [48, 49].

¹⁰<https://github.com/understandable-machine-intelligence-lab/Quantus>

¹¹https://github.com/aimagelab/ADCC?fbclid=IwAR0YK_931xp4pZQnt34S1A9aeNCLRX8m0u8yTPxbTXi80qiyhTiqxWaQ7o

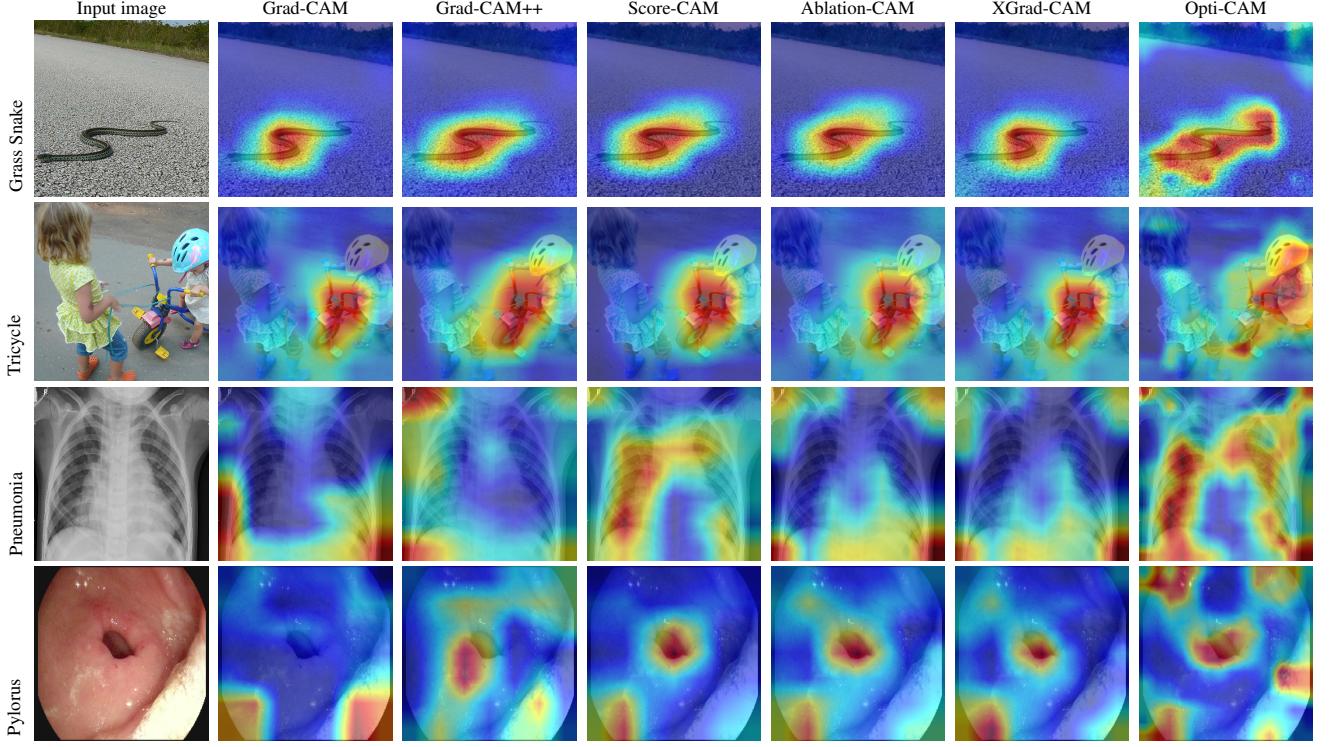


Fig. 2. Saliency maps obtained by different methods for ImageNet (top two rows), Chest X-ray (row 3) and Kvasir (row 4) with VGG. Ground truth class shown on the left of the input image.

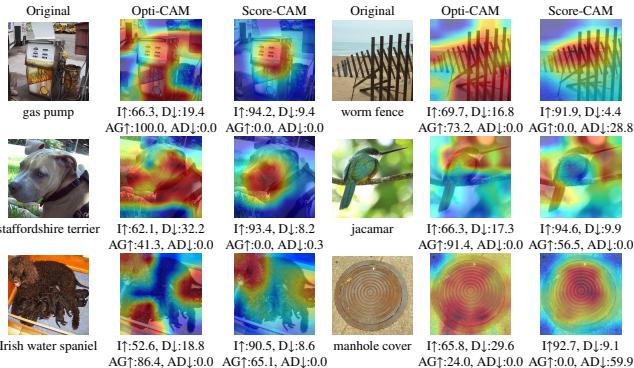


Fig. 3. Failure examples of Opti-CAM regarding insertion/deletion.

To investigate the relative importance of the object and its context, we measure classification metrics when using the bounding box B itself as a saliency map as well as its complement $I \setminus B$, where I is the image. We also evaluate the intersection $B \cap S$ of the saliency map S with the bounding box and with its complement ($S \setminus B$).

As shown in Table 5, the ground truth region of the object is not the only one responsible for the network decision. For example, the bounding box fails both when used as a saliency map itself and when combined with any saliency map, by harming all classification metrics. Even the complement is more effective than the bounding box itself, either alone or when combined. These findings support the hypothesis that localization metrics based on the ground truth bounding box are not necessarily appropriate for evaluating explanations of network decisions. Classification metrics are clearly more appropriate in this sense.

Nevertheless, we report localization metrics in the supplementary material. In summary, although its saliency maps are more spread out, Opti-CAM outperforms other methods on a number of metrics.

5.5. Ablation study

We perform an ablation study of different choices of the objective function (10) and normalization (4) of the saliency map. More choices of (10), layer ℓ , number of iterations and learning rates, selector function g_c and initialization of \mathbf{w} are studied in the supplementary material.

Table 4. $AOPC/MS/ADCC$ scores on ImageNet validation set; \downarrow/\uparrow : lower / higher is better.

METHOD	RESNET50			VGG16		
	$AOPC \uparrow$	$MS \downarrow$	$ADCC \downarrow$	$AOPC \uparrow$	$MS \downarrow$	$ADCC \downarrow$
Grad-CAM [15]	11.7	1.05	74.3	13.1	1.10	73.7
Grad-CAM++ [16]	11.6	1.04	73.6	11.6	1.09	74.6
Score-CAM [13]	10.2	1.04	61.0	11.0	1.09	73.9
XGrad-CAM [37]	11.9	1.05	74.3	13.1	1.10	73.9
Ablation-CAM [12]	11.1	1.04	71.5	12.5	1.10	75.5
Layer-CAM [60]	13.0	1.22	61.1	13.3	1.25	51.7
ExPerturbation [19]	12.0	1.07	26.0	11.2	1.09	42.8
Opti-CAM (ours)	6.3	1.03	65.5	8.9	1.06	70.0

METHOD	AD↓			AG↑			AI↑		
	S	B ∩ S	S \ B	S	B ∩ S	S \ B	S	B ∩ S	S \ B
$S := B$	67.2	—	—	2.3	—	—	9.2	—	—
$S := I \setminus B$	44.0	—	—	2.8	—	—	16.3	—	—
Fake-CAM [21]	0.5	67.2	44.1 0.7	2.3	2.8	42.0	9.2	18.9	
Grad-CAM [15]	15.0	72.6	52.1 15.3	1.8	6.0	40.4	8.4	19.4	
Grad-CAM++ [16]	16.5	72.9	53.1 10.6	1.6	4.1	35.2	7.3	17.1	
Score-CAM [13]	12.5	71.5	50.5 16.1	2.2	6.3	42.5	8.6	20.8	
Ablation-CAM [12]	15.1	72.8	52.1 13.5	1.7	5.6	39.9	7.8	19.0	
XGrad-CAM [37]	14.3	72.6	51.4 15.1	1.8	6.0	42.1	8.0	20.1	
Layer-CAM [60]	49.2	84.2	74.4 2.7	0.4	1.2	12.7	4.4	7.3	
ExPerturbation [19]	43.8	81.6	71.0 7.1	1.4	3.2	18.9	5.6	11.1	
Opti-CAM (ours)	1.4	62.5	34.8 66.3	8.7	25.8	92.5	18.6	47.1	

Table 5. *Bounding box study. Classification metrics on ImageNet validation set using VGG16. B: ground-truth box used by localization metrics; I: entire image; S: saliency map. AD/AI: average drop/increase [16]; AG: average gain (ours); ↓ / ↑: lower / higher is better; bold: best, excluding Fake-CAM.*

Normalization function. For normalization function n (10), we investigate three choices:

$$\text{range : } n(A) := \frac{A - \min A}{\max A - \min A} \quad (16)$$

$$\text{maximum : } n(A) := \frac{A}{\max A} \quad (17)$$

$$\text{sigmoid : } n(a_{ij}) := \frac{1}{1 + e^{-a_{ij}}}, \quad (18)$$

where a_{ij} is element (i, j) of matrix A . The default is (16), normalizing by the range of values in the saliency map, as in Score-CAM (4); while (17) normalizes by the maximum value and (18) by the sigmoid function element-wise.

Objective function. We refer to the default definition of F_ℓ^c (10) as Mask because it maximizes the logit for the masked image. We also consider an alternative definition of objective function F_ℓ^c , which encourages the masked version to preserve the prediction of original image:

$$F_\ell^c(\mathbf{x}; \mathbf{u}) := -|g_c(f(\mathbf{x})) - g_c(f(\mathbf{x} \odot n(\text{up}(S_\ell(\mathbf{x}; \mathbf{u})))))|. \quad (19)$$

This function is named Diff as it minimizes the difference of logits between the masked and the original image.

Results. Table 6 shows classification metrics for the different choices of Opti-CAM, as well as comparison to other methods for reference, for the small subset of ImageNet validation set.

We observe that the choice of normalization function has little effect overall and Sigmoid offers lower performance. Note that the minimum value of saliency maps is often zero or close to zero: Saliency maps are non-negative as convex combinations of non-negative feature maps (8). By contrast, the choice of loss function has more impact on performance and we observe that Mask (10) is superior on all cases.

6. Discussion and conclusions

Opti-CAM combines ideas of different saliency map generation methods, which are masking-based and CAM-based. Our method optimizes the saliency map at inference given a single input image. It does not require any additional data or training any other network, which would need interpretation too.

METHOD	F_ℓ^c	n	AD↓	AG↑	AI↑
Fake-CAM [21]			0.5	0.7	42.1
Grad-CAM [15]			15.0	15.3	40.4
Grad-CAM++ [16]			16.5	35.2	
Score-CAM [13]			12.5	16.1	42.6
Ablation-CAM [12]			15.1	13.5	39.9
XGrad-CAM [37]			14.3	15.1	42.1
Layer-CAM [60]			49.2	2.7	12.7
ExPerturbation [19]			43.8	7.1	18.9
Opti-CAM (ours)	Mask (10)	Range (16)	1.4	66.3	92.5
	Diff (19)	Range (16)	7.1	18.5	54.9
Opti-CAM (ours)	Mask (10)	Max (17)	1.6	66.2	90.3
	Diff (19)	Max (17)	6.8	17.8	54.5
Opti-CAM (ours)	Mask (10)	Sigmoid (18)	5.0	18.3	57.5
	Diff (19)	Sigmoid (18)	6.5	10.0	45.3

Table 6. *Ablation study using VGG16 on 1000 images of ImageNet validation set. AD/AI: average drop/increase [16]; AG: average gain (ours); ↓ / ↑: lower / higher is better; bold: best, excluding Fake-CAM.*

While Opti-CAM crafts a saliency map in the image space, it does not need any regularization. This is because the saliency map is expressed as a convex combination of feature maps and we only optimize one vector over the feature dimensions. The underlying assumption is that of all CAM-based methods: feature maps contain activations at all regions that are of interest for the classes that are present. Opti-CAM is more expensive than non-iterative gradient-based methods but as fast or faster than gradient-free methods that require as many forward passes as channels.

We find that Opti-CAM brings impressive performance improvement over the state of the art according to the most important classification metrics on several datasets. The saliency maps are more spread out compared with those of the competition, attending to larger parts of the object, multiple instances and background context, which may be helpful in classification.

Our new classification metric AG aims to be paired AD as a replacement of AI and resolves a long-standing problem in evaluating attribution methods, without further increasing the number of metrics. We provide strong evidence supporting that the use of ground-truth object bounding boxes for localization is not necessarily optimal in evaluating the quality of a saliency map, because the primary objective is to explain how a classifier works.

Acknowledgements

This publication has received funding from the Excellence Initiative of Aix-Marseille Université - A*Midex, a French “Investissements d’Avenir programme” (AMX-21-IET-017), and the UnLIR ANR project (ANR-19-CE23-0009). Part of this work was performed using HPC resources from GENCI-IDRIS (Grant 2020-AD011013110).

Appendices

Implementation details are provided in Appendix A. In Appendix B, we define localization metrics and provide corresponding results. We provide results on saliency maps of other classes in Appendix C and subjective evaluation in Appendix

D. We provide results on medical data in Appendix E. We then provide more ablation results in Appendix F, sanity check in Appendix G, and results without input image normalization in Appendix H.

Appendix A. Implementation details

All input images are resized to $224 \times 224 \times 3$. To optimize the saliency map with Opti-CAM (9), we use the Adam [50] optimizer with learning rate 0.1 by default, setting the maximum number of iterations to 100 and stopping early when the change in loss is less than 10^{-10} . For VGG16, we generate the saliency map (8) from the feature maps of the last convolutional layer before max pooling by default, *i.e.* convolutional layer 3 of block 5. For ResNet50, we choose the last convolutional layer by default, *i.e.* convolutional layer 3 of bottleneck 2 of block 4. For ViT and DeiT, we choose the last self-attention block by default, *i.e.* layer normalization of self-attention block 12. Ablations concerning the layer ℓ and the convergence of Opti-CAM are included in Appendix F.

Appendix B. Localization metrics

Several works measure the localization ability of saliency maps, using metrics from the *weakly-supervised object localization* (WSOL) task. While we show in the main paper that localization of the object and classifier interpretability are not well aligned as tasks, we still provide localization results here. We use the *official metric* (OM), *localization error* (LE), *pixel-wise F₁ score*, *box accuracy* (BoxAcc) [66], standard pointing game (SP) [30], *energy pointing game* (EP) [13] and *saliency metric* (SM) [17] on the ILSVRC2014¹² dataset. The goal of these metrics is to compare the saliency maps with bounding boxes around the object of interest. For simplicity, we define these metrics for a single image; the reported results are averaged over all images of the test set.

Appendix B.1. Definitions

We are given the saliency map S^c obtained from test image \mathbf{x} for ground truth class c . We denote by $S_{\mathbf{p}}^c$ its value at pixel \mathbf{p} . We binarize the saliency map by thresholding at its average value and we take the bounding box of the largest connected component of the resulting mask as the predicted bounding box B_p , represented as a set of pixels. We compare this box against the set of ground truth bounding boxes \mathcal{B} , which typically contains 1 or 2 boxes of the same class c , or with their union $U = \cup \mathcal{B}$, again represented as a set of pixels. We also compare the predicted class label c_p with the ground truth label c . All metrics take values in $[0, 1]$ and are expressed as percentages, except SM (A7), which is unbounded.

Official Metric (OM). measures the maximum overlap of the predicted bounding box with any ground truth bounding box, requiring that the predicted class label is correct:

$$\text{OM} := 1 - \left(\max_{B \in \mathcal{B}} \text{IoU}(B, B_p) \right) \mathbb{1}_{c_p=c}, \quad (\text{A1})$$

where IoU is intersection over union.

Localization Error (LE). is similar but ignores the predicted class label:

$$\text{LE} := 1 - \max_{B \in \mathcal{B}} \text{IoU}(B, B_p). \quad (\text{A2})$$

Pixel-wise F₁ score (F1). is defined as $F_1 = 2 \frac{PR}{P+R}$, where *precision P* is the fraction of mass of the saliency map that is within the ground truth union

$$P := \frac{\sum_{\mathbf{p} \in U} S_{\mathbf{p}}^c}{\sum_{\mathbf{p}} S_{\mathbf{p}}^c} \quad (\text{A3})$$

and *recall R* is the fraction of the ground truth union that is covered by the saliency map

$$R := \frac{\sum_{\mathbf{p} \in U} S_{\mathbf{p}}^c}{|U|}. \quad (\text{A4})$$

Box Accuracy (BA) [66]. Given threshold values η and δ , we find the bounding box B_p^η of the largest connected component of the binary mask $\{\mathbf{p} : S_{\mathbf{p}} > \eta\}$ and require that it overlaps by δ with at least one ground truth box:

$$\text{BoxAcc}(\eta, \delta) := \max_{B \in \mathcal{B}} \mathbb{1}_{\text{IoU}(B_p^\eta, B) \geq \delta}. \quad (\text{A5})$$

After averaging over the test images, we take the maximum of this measure over a set of values η and then the average over a set of values δ .

Standard Pointing game (SP) [30]. We find the pixel $\mathbf{p}^* := \arg \max_{\mathbf{p}} S_{\mathbf{p}}^c$ having the maximum saliency value and require that it lands in any of the ground truth bounding boxes:

$$\text{SP} := \mathbb{1}_{\mathbf{p}^* \in U}. \quad (\text{A6})$$

Energy Pointing game (EP) [13]. is equivalent to precision (A3).

Saliency Metric (SM) [17]. penalizes the size of the predicted bounding box B_p relative to the image and the cross-entropy loss:

$$\text{SM} := \log \max \left(0.05, \frac{|B_p|}{hw} \right) - \log p^c, \quad (\text{A7})$$

where $h \times w$ is the input image resolution and p^c is the predicted probability for ground truth class label c .

¹²<https://www.image-net.org/challenges/LSVRC/2014/index#>

METHOD	OM↓	LE↓	F1↑ BA↑	SP↑	EP↑	SM↓
RESNET50						
Fake-CAM [21]	63.6	54.0	57.7 47.9	99.8	28.5	0.98
Grad-CAM [15]	72.9	65.8	49.8 56.2	69.8	33.3	1.30
Grad-CAM++ [16]	73.1	66.1	50.4 56.2	69.9	33.1	1.29
Score-CAM [13]	72.2	64.9	49.6 54.5	68.7	32.4	1.25
Ablation-CAM [12]	72.8	65.7	50.2 56.1	69.9	33.1	1.26
XGrad-CAM [37]	72.9	65.8	49.8 56.2	69.8	33.3	1.30
Layer-CAM [60]	73.1	66.0	50.1 55.5	70.0	33.0	1.29
ExPerturbation [19]	73.6	66.6	37.5 44.2	64.8	38.2	1.59
Opti-CAM (ours)	72.2	64.8	47.3 49.2	59.4	30.5	1.34
VGG16						
Fake-CAM [21]	64.7	54.0	57.7 47.9	99.8	28.5	1.07
Grad-CAM [15]	71.1	62.3	42.0 54.2	64.8	32.0	1.39
Grad-CAM++ [16]	70.8	61.9	44.3 55.2	66.2	32.3	1.38
Score-CAM [13]	71.2	62.5	45.3 58.5	68.2	33.4	1.40
Ablation-CAM [12]	71.3	62.6	43.2 56.2	65.7	32.7	1.39
XGrad-CAM [37]	70.8	62.0	41.9 53.5	64.4	31.6	1.41
Layer-CAM [60]	70.5	61.5	28.0 54.7	65.0	32.4	1.45
ExPerturbation [19]	74.1	66.4	37.8 43.3	62.7	36.1	1.74
Opti-CAM (ours)	69.1	59.9	44.1 51.2	61.4	30.7	1.34

Table A7. Localization metrics on ImageNet validation set. OM: official metric; LE: localization error; F1: pixel-wise F1 score; BA: box accuracy; SP: standard pointing game; EP: energy pointing game; SM: saliency metric. ↓ / ↑: lower / higher is better. Bold: best, excluding Fake-CAM.

Appendix B.2. Results

We evaluate the localization ability of saliency maps obtained by our Opti-CAM and we compare with other attribution methods quantitatively. Table A7 and Table A8 report localization metrics on ImageNet. We observe different behavior in different metrics. In particular, Opti-CAM on ResNet and VGG performs best on OM and LE but poorly on the remaining metrics. On transformers, Opti-CAM performs best on OM, LE, F1, and SM.

Metrics, where Opti-CAM does not perform well, are mostly the ones that penalize saliency maps that are more spread out. For example, SP and EP penalize saliency outside the ground truth bounding box of an object. This is not necessarily a weakness of Opti-CAM, because rather than weakly supervised object localization, the objective here is to explain how the classifier works.

Appendix C. Saliency Map of Other Class.

We generate saliency maps for the most probable, second most probable, and least probable classes across the entire dataset using our method and baseline methods such as Grad-CAM, Grad-CAM++, and Score-CAM. These saliency maps are evaluated using the metrics AG, AI, and AD relative to the ground truth. Additionally, we specifically present the results for the most probable classes only on misclassified images in the validation set.

As shown in Table A9, when replacing the ground truth with the predicted (most probable) class, the performance of all methods slightly drops. The results for the second most probable and the least probable class are even lower and similar with each other. In each case, the performance of Opti-CAM suffers more than the others. This result is positive for Opti-CAM because most examples are correctly classified so that the most probable class is the ground truth (*e.g.* accuracy 76.1%) and

METHOD	OM↓	LE↓	F1↑ BA↑	SP↑	EP↑	SM↓
ViT-B						
Fake-CAM [21]	62.8	54.0	57.7 47.9	99.8	28.6	0.87
Grad-CAM [15]	79.6	74.3	29.4 45.0	58.1	31.0	3.27
Grad-CAM++ [16]	84.2	80.6	14.8 23.8	51.4	27.3	4.15
Score-CAM [13]	77.6	71.6	46.0 54.3	66.1	33.1	3.14
XGrad-CAM [37]	82.0	76.9	19.6 41.3	52.8	28.5	3.31
Layer-CAM [60]	70.7	63.9	20.6 50.5	60.7	32.6	1.44
ExPerturbation [19]	71.5	64.9	35.9 44.6	62.3	35.3	1.34
RawAtt [58]	72.4	64.8	18.5 50.4	55.4	31.6	1.68
Rollout [62]	67.6	58.8	36.9 50.7	57.8	30.0	1.16
TIBAV [63]	70.1	63.1	26.6 58.8	66.1	35.0	1.23
Opti-CAM (ours)	64.4	54.6	54.5 48.0	58.2	28.7	0.98
DeiT-B						
Fake-CAM [21]	61.4	54.0	57.7 47.9	99.8	28.7	0.83
Grad-CAM [15]	65.5	60.3	44.3 47.2	62.8	30.2	1.20
Grad-CAM++ [16]	70.6	67.2	34.3 43.6	57.7	30.3	2.14
Score-CAM [13]	79.9	76.2	31.9 43.8	63.4	32.2	3.14
XGrad-CAM [37]	82.0	78.4	19.5 44.1	53.4	28.8	3.03
Layer-CAM [60]	80.2	77.3	17.6 50.8	62.7	35.1	3.15
ExPerturbation [19]	69.9	64.3	36.2 44.2	63.1	35.5	1.16
RawAtt [58]	73.5	68.2	5.9 48.1	46.5	27.3	1.91
Rollout [62]	63.9	57.0	27.8 47.9	36.5	27.2	0.94
TIBAV [63]	68.2	62.2	28.1 59.6	64.1	33.5	1.08
Opti-CAM	62.3	55.1	53.9 48.0	55.1	28.8	0.84

Table A8. Localization metrics with ViT and DeiT on ImageNet validation set. OM: official metric; LE: localization error; F1: pixel-wise F1 score; BA: box accuracy; SP: standard pointing game; EP: energy pointing game; SM: saliency metric. ↓ / ↑: lower / higher is better. Bold: best, excluding Fake-CAM.

Opti-CAM obtains a saliency map explaining the background (any class other than the ground truth); so when we mask the input image with this saliency map, the foreground is hidden and its performance should drop.

For misclassified images, the performance of Opti-CAM suffers even more than the others. Again, this is positive for Opti-CAM because it obtains a saliency map explaining the background (again, different than the ground truth), so the performance for the foreground should drop for the same reason as above. This drop is even higher because the probability for the predicted class is higher than for the second or least probable class.

Appendix D. Subjective Evaluation

As discussed in section 2, the question is not how aligned a network is with human behavior, but to find how the network actually behaves. Nevertheless, for completeness, we perform one experiment as requested. In particular, we now conduct a small survey to explore the ability of humans to compare networks based on the saliency maps generated by Opti-CAM. We randomly select 20 correctly classified images from the validation set of ImageNet and generate corresponding saliency maps using Opti-CAM for both ResNet50 and VGG16. We then conceal information about the network and present the saliency maps along with original images to 38 participants, asking them to determine which network performed better, solely based on the saliency maps. The results indicate

Methods	Ground Truth			Most			Second			Least			Most (Mis.)		
	AD↓	AG↑	AI↑	AD↓	AG↑	AI↑	AD↓	AG↑	AI↑	AD↓	AG↑	AI↑	AD↓	AG↑	AI↑
Grad-CAM [15]	12.2	17.6	44.4	16.54	16.2	39.1	96.26	0.0	2.2	96.28	0.0	2.2	37.84	5.4	40.1
Grad-CAM++ [16]	12.9	16.0	42.1	15.81	15.2	38.8	17.08	11.9	34.8	17.18	11.8	34.5	34.49	5.5	42.4
Score-CAM [13]	8.6	26.6	56.7	10.07	26.9	52.3	26.16	8.4	23.3	28.29	7.4	22.1	36.82	5.1	38.8
Opti-CAM	1.5	68.8	92.8	12.01	60.1	78.3	28.45	5.8	18.8	29.17	5.5	18.1	64.45	2.2	16.2

Table A9. Classification metrics on ImageNet validation set, using ResNet50. AD/AI: average drop/increase [16]; AG: average gain (ours); ↓ / ↑: lower / higher is better; Ground Truth: ground truth class; Most: most probable class; Second: second probable class; Least: least probable class; Most (Mis.): most probable class on misclassified images.

that in 58.15% of cases, participants perceived VGG16 as more accurate than ResNet50. Upon querying the participants about their decision-making process, we find a preference for saliency maps containing larger regions of the object.

Appendix E. Medical data

Medical image recognition is a high-stakes task that crucially needs interpretable models. We thus evaluate our method on two standard medical image classification datasets.

Appendix E.1. Datasets

Chest X-ray. [53] aims at recognizing chest images of patients with pneumonia from healthy ones with 5,216 training images, 16 for validation and 624 for testing. Images are resized to $224 \times 224 \times 3$ to adapt to the pretrained models.

Kvasir. [54] contains 8 classes and aims at recognizing anatomical landmarks, pathological findings and endoscopic procedures inside the gastrointestinal tract. The 8,000 images are split into 6,000 images for training, 1,000 for validation and 1,000 for testing. Images are resized as for the other datasets

Appendix E.2. Network fine-tuning

To train our models on the medical data, we first train the last fully-connected layer according to the classes in each dataset, while keeping the backbone frozen. On Chest X-ray, we use learning rate 10^{-3} for both networks. On Kvasir, we use learning rate 10^{-4} for ResNet50 and 5×10^{-3} for VGG16. We then fine-tune the entire network with a learning rate 10^{-5} for 50 epochs, using SGD with momentum 0.9 for both networks on both datasets. On Chest X-ray data, we obtain accuracies of 83.2% for VGG16 and 82.0% for ResNet50; on Kvasir, 89.5% for VGG16 and 89.8% for ResNet50.

Appendix E.3. Results

Table A10 reports metrics AD/AG/AI and Table A11 reports metrics I/D on Chest X-ray and Kvasir using RESNET50 and VGG16 networks. The conclusions remain the same as for ImageNet. Opti-CAM achieves an average performance on I/D and performs best D on VGG16 of KVASIR. More than that, AD and AI are near perfect in most cases and AG is also extremely high. Additional visualizations are presented in supplementary material.

METHOD	RESNET50			VGG16		
	AD↓	AG↑	AI↑	AD↓	AG↑	AI↑
CHEST X-RAY						
Fake-CAM [21]	0.1	0.9	49.7	0.1	0.4	29.8
Grad-CAM [15]	20.4	29.7	48.7	36.8	39.8	42.3
Grad-CAM++ [16]	24.7	24.1	41.2	36.9	43.4	45.8
Score-CAM [13]	21.6	27.7	44.2	35.3	47.4	48.9
Ablation-CAM [12]	26.2	27.9	42.9	36.9	46.9	47.8
XGrad-CAM [37]	20.4	29.7	48.7	34.7	47.3	50.2
Layer-CAM [60]	24.5	23.4	39.1	36.6	45.9	47.6
ExPerturbation [19]	21.4	5.5	17.9	29.7	21.8	28.7
Opti-CAM (ours)	0.1	91.2	98.4	0.0	85.9	86.2
KVASIR						
Fake-CAM [21]	0.1	0.4	48.3	0.0	0.3	45.0
Grad-CAM [15]	10.0	23.2	39.8	33.8	6.3	14.6
Grad-CAM++ [16]	11.2	18.7	32.9	20.7	9.3	20.4
Score-CAM [13]	9.1	26.7	40.8	8.4	24.0	39.4
Ablation-CAM [12]	10.7	21.6	35.4	10.6	20.9	36.9
XGrad-CAM [37]	10.0	23.2	39.8	12.1	21.6	35.2
Layer-CAM [60]	11.7	18.2	32.5	12.9	17.1	30.8
ExPerturbation [19]	48.4	13.8	21.0	34.8	19.0	27.7
Opti-CAM (ours)	0.2	91.1	99.0	0.0	93.5	98.1

Table A10. Classification metrics on Chest X-ray and KVASIR datasets. AD/AI: average drop/increase [16]; AG: average gain (ours); ↓ / ↑: lower / higher is better; Bold: best, excluding Fake-CAM.

METHOD	CHEST X-RAY				KVASIR			
	RESNET50		VGG16		RESNET50		VGG16	
	I↑	D↓	I↑	D↓	I↑	D↓	I↑	D↓
Grad-CAM [15]	83.0	75.7	85.0	81.9	81.3	32.2	72.1	48.9
Grad-CAM++ [16]	82.2	79.1	85.1	81.8	80.2	33.8	72.1	48.7
Score-CAM [13]	82.9	77.0	87.6	79.0	80.6	33.4	79.3	34.9
Ablation-CAM [12]	83.5	75.1	92.0	73.1	80.3	32.6	79.4	36.2
XGrad-CAM [37]	82.9	75.6	88.7	75.6	81.3	32.2	79.2	36.6
Opti-CAM (ours)	82.0	78.4	86.8	79.5	80.2	37.7	77.0	24.8

Table A11. I/D: insertion/deletion [22] on Chest X-ray and KVASIR dataset using both RESNET50 and VGG16. ↓ / ↑: lower / higher is better.

Appendix F. More ablations

Appendix F.1. Selectivity

We investigate the effect of the selectivity of saliency maps on classification performance. In particular, before evaluation, we raise saliency maps element-wise to an exponent α that takes values in $\{0.01, 0.05, 0.1, 0.5, 1, 1.5, 2, 3, 5, 10\}$. When α is small, the saliency maps become more uniform, so that more information about the original image is revealed to the network. Respectively, when α is large, the saliency maps become more selective, so that the network sees fewer parts of the input. The order of pixels is maintained.

Results in terms of AD, AG, AI are shown in Figure A4, averaged over 1,000 ImageNet images. We observe that AD stays

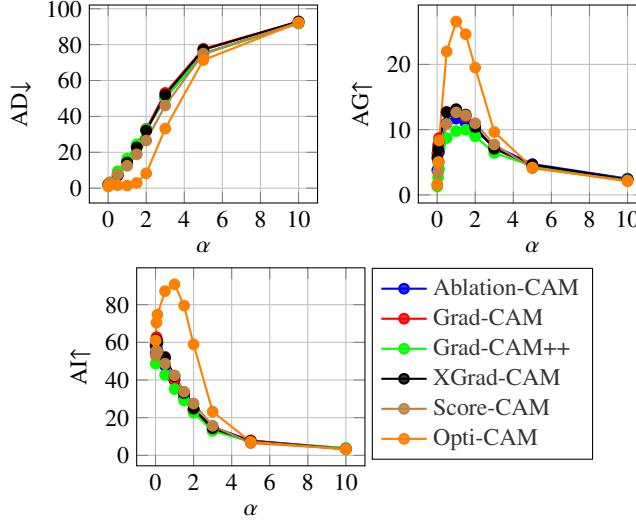


Fig. A4. Effect of selectivity (raising element-wise to exponent α) of saliency maps on classification performance. AD/AI: average drop/increase [16]; AG: average gain (ours); \downarrow / \uparrow : lower / higher is better.

near zero for Opti-CAM for $\alpha < 2$, while it increases linearly with α for the other methods. The AG and AI of Opti-CAM has a strong peak at $\alpha = 1$, i.e. for the original saliency maps. The other methods are less sensitive and their AI performance is not optimal at $\alpha = 1$.

Appendix F.2. Opti-CAM components

Objective function. We consider more alternative definitions of the objective function F_ℓ^c , taking into account not only the regions inside the saliency maps (In) but also their complement, outside (Out). In particular, relative to Mask, we define IOMask as

$$F_\ell^c(\mathbf{x}; \mathbf{u}) := g_c(f(\mathbf{x} \odot \mathbf{s})) - g_c(f(\mathbf{x} \odot (1 - \mathbf{s}))), \quad (\text{A1})$$

where $\mathbf{s} := n(\text{up}(S_\ell(\mathbf{x}; \mathbf{u})))$ for brevity. Similarly, relative to Diff, we define IODiff as

$$\begin{aligned} F_\ell^c(\mathbf{x}; \mathbf{u}) := & -|g_c(f(\mathbf{x})) - g_c(f(\mathbf{x} \odot \mathbf{s}))| \\ & + |g_c(f(\mathbf{x})) - g_c(f(\mathbf{x} \odot (1 - \mathbf{s})))|. \end{aligned} \quad (\text{A2})$$

According to Table A12, IOMask performs great on AD and AI but worse on AG, while IODiff is worse on all metrics. Therefore, including the complementary of the saliency map is not beneficial.

Layers. Table A13 shows how the performance of Opti-CAM, in terms of AD/AI/AG, depends on the layer ℓ of the VGG16 network used to compute the saliency map S_ℓ^c (8). We can see that the layers 26, 29, and 42 are all competitive. We choose the last convolutional layer (42) to be compatible with the other CAM methods [11, 15, 16, 13].

Convergence. Finally, Figure A5 shows the classification performance of Opti-CAM vs. number of iterations for different learning rates. Optimal performance can be obtained at 100 iterations with learning rate $\eta = 0.1$. We use these settings by

METHOD	F_ℓ^c	AD \downarrow	AG \uparrow	AI \uparrow
Fake-CAM [21]		0.5	0.7	42.1
Grad-CAM [15]		15.0	15.3	40.4
Grad-CAM++ [16]		16.5	10.6	35.2
Score-CAM [13]		12.5	16.1	42.6
Ablation-CAM [12]		15.1	13.5	39.9
XGrad-CAM [37]		14.3	15.1	42.1
Layer-CAM [60]		49.2	2.7	12.7
ExPerturbation [19]		43.8	7.1	18.9
<hr/>				
Opti-CAM		Mask (10)	1.4	66.3
		Diff (19)	7.1	18.5
		IOMask (A1)	0.2	5.5
		IODiff (A2)	25.9	7.6
<hr/>				

Table A12. Ablation study on objective function using VGG16 on 1000 images of ImageNet validation set. Choices for objective function F_ℓ^c : Mask: (10); Diff: (19); IOMask: (A1); IODiff: (A2). Choice for normalization function n : Range (16). Iterations: 50. AD/AI: average drop/increase [16]; AG: average gain (ours); \downarrow / \uparrow : lower / higher is better.

LAYER	AD \downarrow	AG \uparrow	AI \uparrow	LAYER	AD \downarrow	AG \uparrow	AI \uparrow
42	1.4	66.0	92.5	36	1.7	66.1	90.3
32	2.8	61.3	81.6	29	1.6	78.0	93.9
26	1.7	80.1	93.7	22	3.3	68.8	84.8
19	2.9	67.3	84.9	16	2.3	72.4	89.1
12	4.1	61.9	82.4	9	4.3	44.2	71.9
6	13.5	23.5	50.2				

Table A13. Layer ablation on 1,000 images from ImageNet validation set, using various layers of VGG16. The last convolutional layer before max pooling is chosen as our default layer (layer 42). AD/AI: average drop/increase [16]; AG: average gain (ours); \downarrow / \uparrow : lower / higher is better.

default. We note that by using 50 iterations allows us to double the speed at the cost of a 6% drop of AG and very small drop of AI and AD.

Appendix G. Sanity check

We use the model parameter randomization test proposed by [67]. This test compares the saliency maps generated by a trained model with the ones generated by a partially randomly initialized network of the same architecture. In particular, we choose 5 layers of ResNet50 and we progressively replace them by random ones so that we have 6 different models with different amount of random parameters. The saliency maps are generated for the small subset of ImageNet validation set, as in the ablation study.

Following [67], we compute a number of similarity metrics between these saliency maps generated by the original and the randomized network, including Rank Correlation with/without absolute values, HOGs similarity, and SSIM. The results are shown in Figure A6 (saliency map similarity measurements) and Figure A7 (saliency map visualizations). Our method passes the sanity check, as it is very sensitive to changes in the model parameters. We also use model parameter randomization test and train a ResNet50 with randomly permuted labels following the training recipes from the pytorch models¹³. The SSIM similarity is 0.013, which shows that Opti-CAM is sensitive to the relationship between instances and labels.

¹³<https://github.com/pytorch/vision/tree/main/references/classification>

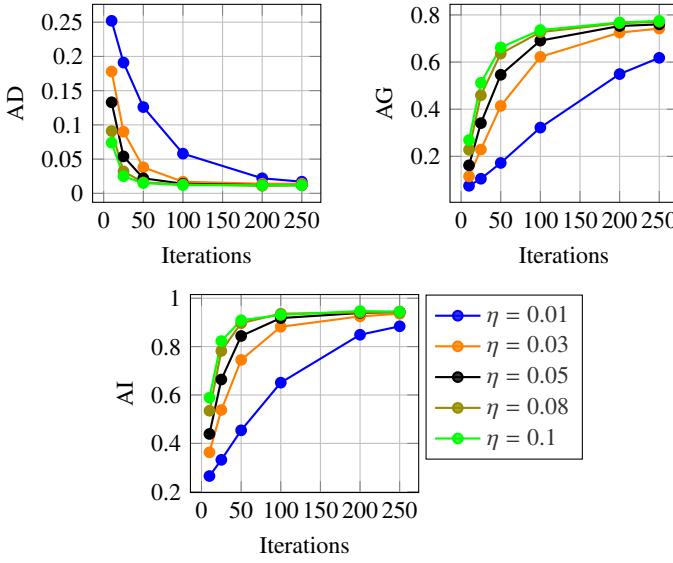


Fig. A5. Classification metrics vs. number of iterations for different learning rates, using VGG-16 on 1000 images of ImageNet. AD/AI: average drop/increase [16]; AG: average gain (ours); ↓ / ↑: lower / higher is better.

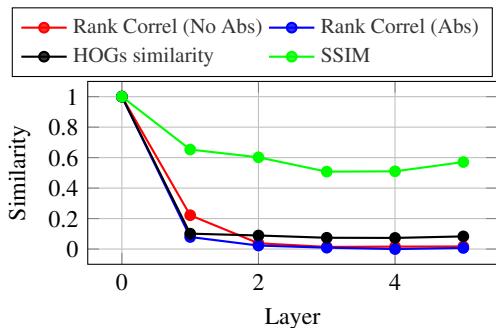


Fig. A6. Sanity check of Opti-CAM on 1,000 images of ImageNet validation set using ResNet50. Similarity between saliency maps by original and randomized network, where layers are progressively replaced by random ones.

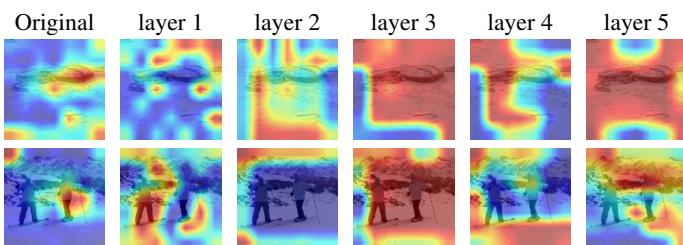


Fig. A7. Sanity check visualization of Opti-CAM on two images of ImageNet validation set using ResNet50. First column: Opti-CAM saliency maps for the original network; remaining columns: Opti-CAM saliency maps where layers are progressively replaced by random ones.

METHOD	RESNET50			VGG16				
	AD↓	AG↑	AI↑	T	AD↓	AG↑	AI↑	T
Fake-CAM [21]	0.9	0.7	47.4	0.00	0.5	0.3	47.7	0.00
Grad-CAM [15]	36.4	5.5	27.0	0.03	41.6	3.3	25.2	0.02
Grad-CAM++ [16]	37.6	4.9	24.0	0.04	46.3	2.0	19.0	0.02
Score-CAM [13]	28.8	8.8	33.6	20.47	39.3	3.5	24.6	3.08
Ablation-CAM [12]	36.6	5.1	25.6	18.49	41.8	2.9	24.0	2.95
XGrad-CAM [37]	36.4	5.5	27.0	0.03	40.6	3.4	25.8	0.02
Layer-CAM [60]	42.6	4.2	19.2	0.02	82.1	0.3	6.9	0.01
ExPerturbation [19]	51.2	6.9	26.1	15.67	50.1	4.4	24.5	9.10
Opti-CAM (ours)	2.0	49.4	91.2	3.94	1.5	52.7	92.1	3.95

Table A14. Classification metrics on ImageNet validation set, without input normalization. AD/AI: average drop/increase [16]; AG: average gain (ours); ↓ / ↑: lower / higher is better. T: Average time (sec) per batch of 8 images. **Bold:** best, excluding Fake-CAM.

Appendix H. Results without input normalization

It is standard that images are normalized to zero mean and unit standard deviation before feeding them to a network, because this is how networks are trained. For example, for ImageNet images, we subtract the mean vector [0.485, 0.456, 0.406] and divide channel-wise by standard deviation [0.229, 0.224, 0.225]. By doing so however, we cannot reproduce the results published for several baseline methods; rather, all results are improved dramatically. We can obtain results similar to published ones by *not* normalizing, thus we speculate that authors of related work do not normalize images. This is also suggested by our attempts to communicate with the authors.

We believe normalization is important and we include it in all our experiments. For reference and to allow for comparison with published results, we provide results without normalization in Table A14 that correspond to Table 1. Finally, code is provided to allow for the reproduction and verification of our results.

References

- [1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2018).
- [2] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digital signal processing 73 (2018) 1–15.
- [3] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, K.-R. Müller, Explaining deep neural networks and beyond: A review of methods and applications, Proc. of the IEEE 109 (3) (2021) 247–278.
- [4] F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, S. Rinzivillo, Benchmarking and survey of explanation methods for black box models, CoRR abs/2102.13076 (2021). arXiv:2102.13076.
- [5] X. Li, H. Xiong, X. Li, X. Wu, X. Zhang, J. Liu, J. Bian, D. Dou, Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond, arXiv preprint arXiv:2103.10689 (2021).
- [6] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, ICLR Workshop (2014).
- [7] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, H. Lipson, Understanding neural networks through deep visualization, CoRR abs/1506.06579 (2015). arXiv:1506.06579.
- [8] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: ECCV, 2014.
- [9] J. T. Springenberg, A. Dosovitskiy, T. Brox, M. A. Riedmiller, Striving for simplicity: The all convolutional net, ICLR (2015).

- [10] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, *PloS one* (2015).
- [11] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: *CVPR*, 2016.
- [12] s. desai, H. G. Ramaswamy, Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization, in: *WACV*, 2020.
- [13] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, X. Hu, Score-CAM: Score-weighted visual explanations for convolutional neural networks, in: *CVPR Workshop*, 2020.
- [14] M. B. Muhammad, M. Yeasin, Eigen-cam: Class activation map using principal components, in: *IJCNN*, 2020.
- [15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: *CVPR*, 2017.
- [16] A. Chattopadhyay, A. Sarkar, P. Howlader, V. N. Balasubramanian, Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks, in: *WACV*, 2018.
- [17] P. Dabkowski, Y. Gal, Real time image saliency for black box classifiers, *NIPS* (2017).
- [18] R. C. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in: *ICCV*, 2017.
- [19] R. Fong, M. Patrick, A. Vedaldi, Understanding deep networks via extremal perturbations and smooth masks, in: *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2950–2958.
- [20] K. Schulz, L. Sixt, F. Tombari, T. Landgraf, Restricting the flow: Information bottlenecks for attribution, *arXiv preprint arXiv:2001.00396* (2020).
- [21] S. Poppi, M. Cornia, L. Baraldi, R. Cucchiara, Revisiting the evaluation of class activation mapping for explainability: A novel metric and experimental analysis, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2299–2304.
- [22] V. Petsiuk, A. Das, K. Saenko, Rise: Randomized input sampling for explanation of black-box models, *BMVC* (2018).
- [23] Z. C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery., *Queue* 16 (3) (2018).
- [24] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, in: *SIGKDD, KDD '16*, 2016.
- [25] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), *NIPS*, 2017.
- [26] P. Linardatos, V. Papastefanopoulos, S. Kotsiantis, Explainable ai: A review of machine learning interpretability methods, *Entropy* 23 (1) (2020) 18.
- [27] J. Adebayo, J. Gilmer, I. J. Goodfellow, B. Kim, Local explanation methods for deep neural networks lack sensitivity to parameter values, *ICLR Workshop* (2018).
- [28] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, K. Müller, How to explain individual classification decisions, *J. MLR* (2010).
- [29] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: *ICML*, 2017.
- [30] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, S. Sclaroff, Top-down neural attention by excitation backprop, *International Journal of Computer Vision (IJCV)* 126 (10) (2018) 1084–1102.
- [31] J. Basting, K. Filippova, The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?, *EMNLP Workshop* (2020).
- [32] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, M. Wattenberg, Smoothgrad: removing noise by adding noise, *CoRR abs/1706.03825* (2017). *arXiv:1706.03825*.
- [33] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: *ICML*, 2017.
- [34] S.-A. Rebuffi, R. Fong, X. Ji, A. Vedaldi, There and back again: Revisiting backpropagation saliency methods, in: *CVPR*, 2020.
- [35] A. Elliott, S. Law, C. Russell, Explaining classifiers using adversarial perturbations on the perceptual ball, in: *CVPR*, 2021.
- [36] M. A. A. K. Jalwana, N. Akhtar, M. Bennamoun, A. Mian, Attack to explain deep representation, in: *CVPR*, 2020.
- [37] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, B. Li, Axiom-based grad-cam: Towards accurate visualization and explanation of cnns, *BMVC* (2020).
- [38] M. He, B. Li, S. Sun, A survey of class activation mapping for the interpretation of convolution neural networks, in: *International Conference On Signal And Information Processing, Networking And Computers*, Springer, 2022, pp. 399–407.
- [39] C. Chang, E. Creager, A. Goldenberg, D. Duvenaud, Explaining image classifiers by counterfactual generation, *ICLR* (2019).
- [40] J. Phang, J. Park, K. J. Geras, Investigating and simplifying masking-based saliency methods for model interpretability, *arXiv preprint arXiv:2010.09750* (2020).
- [41] K. Zolna, K. J. Geras, K. Cho, Classifier-agnostic saliency map extraction, *CVIU* 196 (2020) 102969.
- [42] H. Bilen, A. Vedaldi, Weakly supervised deep detection networks, in: *CVPR*, 2016.
- [43] A. Kolesnikov, C. H. Lampert, Seed, expand and constrain: Three principles for weakly-supervised image segmentation, in: *ECCV*, 2016.
- [44] J. Ahn, S. Cho, S. Kwak, Weakly supervised learning of instance segmentation with inter-pixel relations, in: *CVPR*, 2019.
- [45] K. Szczepankiewicz, A. Popowicz, K. Charkiewicz, K. Nalecz-Charkiewicz, M. Szczepankiewicz, S. Lasota, P. Zawistowski, K. Radlak, Ground truth based comparison of saliency maps algorithms, *Scientific Reports* 13 (1) (2023) 16887.
- [46] A. Hedström, L. Weber, D. Krakowczyk, D. Bareeva, F. Motzkus, W. Samek, S. Lapuschkin, M. M. Höhne, Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond, *Journal of Machine Learning Research* 24 (34) (2023) 1–11. URL <http://jmlr.org/papers/v24/22-0142.html>
- [47] T. Gomez, T. Fréour, H. Mouchère, Metrics for saliency map evaluation of deep learning explanation methods, in: *International Conference on Pattern Recognition and Artificial Intelligence*, Springer, 2022, pp. 84–95.
- [48] R. Shetty, B. Schiele, M. Fritz, Not using the car to see the sidewalk—quantifying and controlling the effects of context in classification and segmentation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8218–8226.
- [49] S. Rao, M. Böhle, B. Schiele, Towards better understanding attribution methods, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10223–10232.
- [50] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *ICLR* (2015).
- [51] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *NIPS* (2012).
- [52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, *IJCV* 115 (3) (2015) 211–252.
- [53] D. Kermany, K. Zhang, M. Goldbaum, et al., Labeled optical coherence tomography (oct) and chest x-ray images for classification, *Mendeley data* 2 (2) (2018).
- [54] K. Pogorelov, K. R. Randel, C. Griwodz, S. L. Eskeland, T. de Lange, D. Johansen, C. Spampinato, D.-T. Dang-Nguyen, M. Lux, P. T. Schmidt, et al., Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection, in: *Multimedia Systems Conf.*, 2017.
- [55] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *CVPR*, 2016.
- [56] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *ICLR* (2015).
- [57] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: *ICML*, 2015.
- [58] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, *arXiv preprint arXiv:2010.11929* (2020).
- [59] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jegou, Training data-efficient image transformers & distillation through attention, in: *International Conference on Machine Learning*, Vol. 139, 2021, pp. 10347–10357.
- [60] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, Y. Wei, Layercam: Exploring hierarchical class activation maps for localization, *IEEE Transactions on Image Processing* 30 (2021) 5875–5888.
- [61] R. L. Draelos, L. Carin, Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks, *arXiv preprint arXiv:2011.08891* (2020).
- [62] S. Abnar, W. Zuidema, Quantifying attention flow in transformers, *arXiv preprint arXiv:2005.00928* (2020).

- [63] H. Chefer, S. Gur, L. Wolf, Transformer interpretability beyond attention visualization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 782–791.
- [64] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, Evaluating the visualization of what a deep neural network has learned, *IEEE transactions on neural networks and learning systems* 28 (11) (2016) 2660–2673.
- [65] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, P. K. Ravikumar, On the (in) fidelity and sensitivity of explanations, *Advances in Neural Information Processing Systems* 32 (2019).
- [66] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, H. Shim, Evaluating weakly supervised object localization methods right, in: CVPR, 2020.
- [67] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim, Sanity checks for saliency maps, NIPS (2018).