[1]organization=Centrale Marseille, Aix Marseille Univ, CNRS, LIS, city=Marseille, postcode=13397, country=France

[2]organization=Institute of Advanced Research on Artificial Intelligence (IARAI), city=Vienna, postcode=1030, country=Austria

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.

Compared with gradient-free methods, 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.

We introduce a new evaluation metric, *average gain* (AG), to be paired with *average drop* (AD) as a replacement of *average increase* (AI).

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.

We shed more light into how a classifier may exploit background context.

# Opti-CAM: Optimizing saliency maps for interpretability

Hanwei Zhang[1], Felipe Torres[1], Ronan Sicre[1], Yannis Avrithis[1], Stephane Ayache[1]

---

**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.
*Keywords:* Interpretability; Explainable AI; Saliency map; Class activation maps; Computer vision;

## 1. Introduction

The success of *deep neural networks* (DNN) and their increasing penetration into most sectors of human activity has led to 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 [**? ? ? ? ?** ].

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 [**?** **?** ], including variants of backpropagation [**?** **?** **?** ]. CAM [**?** ] introduced class-specific linear combinations of feature maps, and led to several alternative weighting schemes [**?** **?** **?** ], including the use of gradients [**?** **?** ]. On the other hand, *occlusion-* or *masking-based* methods [**?** **?** **?** **?** ] remove regions in the image space while improving classification performance.

Score-CAM [**?** ] uses each feature map as a mask and defines a corresponding weight based on 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?* The optimal mask should highlight regions for which the network is most confident.

*Masking-based* methods, such as extremal perturbations [**?** ] or IBA [**?** ], do use gradient descent to maximize the class score. The mask is now a variable in the input or feature space and the class score is expressed as a function of the mask directly. 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.

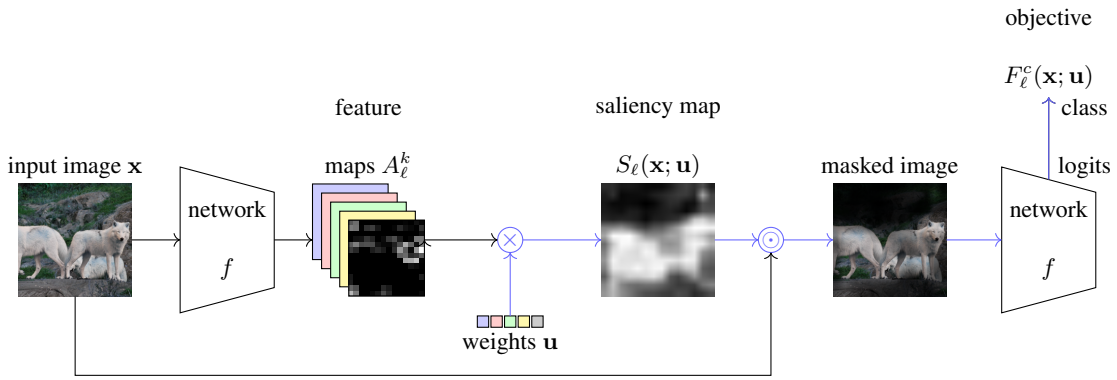


Figure 1: Overview of Opti-CAM. We are given an input image $\mathbf{x}$, a fixed network $f$, a target layer $\ell$ and a class of interest $c$. We extract the feature maps from layer $\ell$ and obtain a saliency map $S_\ell(\mathbf{x};\mathbf{u})$ by forming a convex combination of the feature maps ($\times$) with weights determined by a variable vector $\mathbf{u}$ (**??**). 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 (**??**). We find the value of $\mathbf{u}^*$ that maximizes this logit by optimizing along the path highlighted in blue (**??**), as well as the corresponding optimal saliency map $S_\ell(\mathbf{x};\mathbf{u}^*)$ (**??**).

Motivated by the above, we introduce Opti-CAM, illustrated in **??**. 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 Score-CAM that is optimized iteratively, or 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) [**?** ]. 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 [**?** ] 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 [**?** **?** **?** ], 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) [**?** ].
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 [**?** **?** **?** **?** **?** ]. These works can be categorized into *transparency* and *post-hoc interpretability* [**?** **?** ]. The former addresses how to design an internally understandable model. Here we are interested in the latter, which treats the studied network as a black box and interprets its inner processing [**?** **?** **?** **?** **?** **?** ]. Among post-hoc methods, LIME [**?** ] and SHAP [**?** ] 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 [**?** ], occlusion, or a combination.

*Gradient-based methods.* Gradient-based methods [**?** **?** **?** ] 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 [**?** ], the gradient is directly treated as a saliency map. Inspired by DeconvNet [**?** ], *guided backpropagation* [**?** ] improves the explanation by setting negative gradients to zero using ReLU units. Other methods [**?** **?** **?** ] are inspired by Layer-wise Relevance Propagation (LRP) [**?** ]. SmoothGrad [**?** ] and *integrated gradients* [**?** ] accumulate gradients into saliency maps, while NormGrad [**?** ] attempts to unify gradient-based methods. A different approach is to use adversarial attacks [**?** **?** ]. Several of these methods do not satisfy the fundamental property of implementation invariance [**?** ].

*CAM-based methods.* *Class activation maps* (CAM) [**?** ] 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 [**?** ], GradCAM++ [**?** ], XGradCAM [**?** ], LayerCAM [**?** ], HiRes-CAM [**?** ], and Libra-CAM [**?** ]. Gradient-free methods, including Ablation-CAM [**?** ], Score-CAM [**?** ], SS-CAM [**?** ], F-CAM [**?** ], Abs-CAM [**?** ], Poly-CAM [**?** ] and Shap-CAM [**?** ], rather measure the effect on the target class score of each feature map acting as a mask on the input. 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. RISE [**?** ] randomly masks input images and uses the class score as a weight to define a linear combination. *Meaningful perturbations* [**?** ] and *extremal perturbations* [**?** ] 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) [**?** ] optimizes the mask in the feature space as a tensor instead. Score-CAM [**?** ] 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 [**? ?** ], 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 [**?** ] or auto-encoders [**? ? ?** ]. This approach may be compared with weakly-supervised object detection [**?** ], segmentation [**?** ] or instance segmentation [**?** ]. IBA [**?** ] 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. *Average drop* (AD) and *average increase* (AI), also known as increase in confidence [**?** ] 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) [**?** ] 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 [**? ?** ]. We demonstrate that localization and attribution are not well-aligned as tasks.

## 3. Opti-CAM

### *3.1. Preliminaries*

*Notation.* Consider a classifier network $f : \mathcal{X} \to \mathbb{R}^C$ 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 $p_c = \mathrm{softmax}(\mathbf{y})_c := e^{y_c} / \sum_j e^{y_j}$ the predicted probability for class $c$. For layer $\ell$ with $K_\ell$ channels, we denote by $A^k_\ell = f^k_\ell(\mathbf{x}) \in \mathbb{R}^{h_\ell \times w_\ell}$ the feature map for channel $k \in \{1, \dots, K_\ell\}$, with spatial resolution $h_\ell \times w_\ell$. Because of $\mathrm{relu}$ 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 $\ell$ and a class of interest $c$, we consider saliency maps given by the general formula

$$S^c_\ell(\mathbf{x}) := h\left(\sum_k w^c_k A^k_\ell\right), \tag{1}$$

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

$$w^c_k := \mathrm{GAP}\left(\frac{\partial y_c}{\partial A^k_\ell}\right), \tag{2}$$

where $\mathrm{GAP}$ is global average pooling. The motivation for $\mathrm{relu}$ 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 [**?** ] is also defined for any layer $\ell$ with $h = \mathrm{relu}$ and weights $w_k^c := \mathrm{softmax}(\mathbf{u}^c)_k$. 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 $A_\ell^k$, for all channels $k$:

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

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

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

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

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

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

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

where $\mathbf{w}_0 = \mathbf{0}$, $\delta = 1$ and $\mathbf{e}_k$ is the $k$-th standard basis vector of $\mathbb{R}^{K_\ell}$. This resembles the numerical approximation of the derivative $\frac{\partial F}{\partial w_k}(\mathbf{w}_0)$, except that $\delta$ 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) [**?** ] or *extremal perturbations* [**?** ]. In general, optimization takes the form

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

Here, a mask $\mathbf{m}$ is directly optimized and does not rely on feature maps, hence the saliency map $S^x(\mathbf{x})$ is not connected to any layer $\ell$. 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 $\lambda$. 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 **??**, 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 [**?** ], we use channel weights $w_k := \mathrm{softmax}(\mathbf{u})_k$, where $\mathbf{u} \in \mathbb{R}^{K_\ell}$ is a variable. We then consider saliency map $S_\ell$ in layer $\ell$ as a function of both the input image $\mathbf{x}$ and variable $\mathbf{u}$:

$$S_\ell(\mathbf{x}; \mathbf{u}) := \sum_k \mathrm{softmax}(\mathbf{u})_k A_\ell^k. \tag{8}$$

Comparing with (**??**), $h$ is the identity mapping, because feature maps are non-negative and weights are positive.

*Optimization.* Now, given a layer $\ell$ 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 $S_\ell(\mathbf{x}; \mathbf{u}^*)$:

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

where we define the objective function

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

Here, the saliency map $S_\ell(\mathbf{x}; \mathbf{u})$ is adapted to $\mathbf{x}$ exactly as in (**??**) in terms of resolution and normalization. For *normalization function* $n$, the default is (**??**). 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 $F_\ell^c$ itself, are investigated in **??** and in the supplementary material.

*Opti-CAM.* Putting everything together, we define

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

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

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

*Discussion.* By maximizing (**??**), 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 (**??**), with normalized weights. Hence, the saliency map is discouraged from taking up the entire image, both by the $\mathrm{softmax}$ competition (**??**) and by the fact that feature maps only respond to particular locations.

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

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

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) [**?** ] 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:

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

*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:

$$\mathrm{AI}(\%) := \frac{1}{N}\sum_{i}^{N}\mathbb{1}_{p_i^c < o_i^c}\cdot 100. \tag{14}$$

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, [**?** ] 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 [**?** ] 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:

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

This definition is symmetric to the definition of average drop, in the sense that in absolute value, the numerator in the sum of $\mathrm{AD}, \mathrm{AG}$ 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*

*ImageNet.* We use the validation set of ImageNet ILSVRC 2012 [**?** **?** ], 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 ILSVRC[1] are used on the same validation set.

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

*Networks.* For all datasets, we use the pretrained ResNet50 [**?** ] and VGG16 [**?** ] networks with batch normalization [**?** ] from the Pytorch model zoo[2]. For ImageNet, we further use the pretrained ViT-B (16-224) [**?** ] and DeiT-B (16-224) [**?** ] from Pytorch image models (timm)[3]. Regarding 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) [**?** ] 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) [**?** ] and highlight their limitations. Using classification metrics, we show the limitations of using the localization ground truth for the evaluation of attribution methods. In the

[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

supplementary, we provide a number of localization metrics from the *weakly-supervised object localization* (WSOL) task of ILSVRC2014[4].

*Methods.* We compare against the following state-of-the-art methods: Grad-CAM [**?** ], Grad-CAM++ [**?** ], Score-CAM [**?** ], Ablation-CAM [**?** ], XGrad-CAM [**?** ], Layer-CAM [**?** ], ExtremalPerturbation [**?** ] and HiRes-CAM [**?** ]. Implementations are obtained from the PyTorch CAM library[5] or TorchRay[6]. For transformer models, we also compare against raw attention [**?** ], rollout [**?** ] and TIBAV [**?** ][7].

*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.

*CNN.* **??** shows ImageNet classification metrics using VGG16 and ResNet50. Our Opti-CAM brings impressive 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 [**?** ] 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.

[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/hila-chefer/Transformer-Explainability

| Method | ResNet50 | | | | VGG16 | | | |
|---|---|---|---|---|---|---|---|---|
| | AD↓ | AG↑ | AI↑ | T | AD↓ | AG↑ | AI↑ | T |
| Fake-CAM [**?** ] | 0.8 | 1.6 | 46.0 | 0.00 | 0.5 | 0.6 | 42.6 | 0.00 |
| Grad-CAM [**?** ] | 12.2 | 17.6 | 44.4 | 0.03 | 14.2 | 14.7 | 40.6 | 0.02 |
| Grad-CAM++ [**?** ] | 12.9 | 16.0 | 42.1 | 0.03 | 17.1 | 10.2 | 33.4 | 0.02 |
| Score-CAM [**?** ] | 8.6 | 26.6 | 56.7 | 15.22 | 13.5 | 15.6 | 41.7 | 3.11 |
| Ablation-CAM [**?** ] | 12.5 | 16.4 | 42.8 | 18.26 | 15.5 | 12.6 | 36.9 | 2.98 |
| XGrad-CAM [**?** ] | 12.2 | 17.6 | 44.4 | 0.03 | 13.8 | 14.8 | 41.2 | 0.02 |
| Layer-CAM [**?** ] | 15.6 | 15.0 | 38.8 | 0.08 | 48.9 | 3.1 | 13.5 | 0.07 |
| ExPerturbation [**?** ] | 38.1 | 9.5 | 22.5 | 152.96 | 43.0 | 7.1 | 20.5 | 83.20 |
| HiRes-CAM [**?** ] | 12.2 | 17.6 | 44.4 | 0.03 | 15.8 | 13.2 | 37.8 | 0.02 |
| 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 [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better; T: Average time (sec) per batch of 8 images. Bold: best, excluding Fake-CAM.

In the supplementary material we report *insertion* (I) and *deletion* (D) metrics along with failure cases of Opti-CAM. The latter indicate that our saliency maps are not incorrect as a whole, but capturing more parts of the object, more instances or more background context results in larger or several disconnected salient regions. This does not let the classifier focus on a single discriminative region when pixels are processed sequentially by increasing saliency. Rather, I/D favor smaller and more compact saliency maps.

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

| METHOD | VIT-B | | | | DEIT-B | | | |
|---|---|---|---|---|---|---|---|---|
| | AD↓ | AG↑ | AI↑ | T | AD↓ | AG↑ | AI↑ | T |
| Fake-CAM [**?** ] | 0.3 | 0.4 | 48.3 | 0.00 | 0.6 | 0.3 | 44.6 | 0.00 |
| Grad-CAM [**?** ] | 69.4 | 2.5 | 12.4 | 0.14 | 33.5 | 1.7 | 12.5 | 0.11 |
| Grad-CAM++ [**?** ] | 86.3 | 1.5 | 1.0 | 0.15 | 50.7 | 0.9 | 7.2 | 0.13 |
| Score-CAM [**?** ] | 32.0 | 6.2 | 33.0 | 23.69 | 53.6 | 2.2 | 12.2 | 22.47 |
| XGrad-CAM [**?** ] | 88.1 | 0.4 | 4.3 | 0.13 | 80.5 | 0.3 | 4.1 | 0.12 |
| Layer-CAM [**?** ] | 82.0 | 0.2 | 2.9 | 0.24 | 88.9 | 0.4 | 2.6 | 0.24 |
| ExPerturbation [**?** ] | 28.8 | 6.2 | 24.4 | 133.52 | 60.9 | 2.0 | 8.5 | 129.12 |
| RawAtt [**?** ] | 92.6 | 0.2 | 2.8 | 0.02 | 95.3 | 0.0 | 1.8 | 0.02 |
| Rollout [**?** ] | 42.1 | 5.6 | 20.9 | 0.02 | 55.2 | 0.8 | 7.9 | 0.02 |
| TIBAV [**?** ] | 81.7 | 0.8 | 5.8 | 0.16 | 62.3 | 0.7 | 7.1 | 0.16 |
| HiRes-CAM [**?** ] | 98.4 | 0.0 | 0.7 | 0.03 | 97.2 | 0.0 | 1.2 | 0.03 |
| 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 [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better. T: Average time (sec) per batch of 8 images. Bold: best, excluding Fake-CAM.

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.* **??** 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.

### *5.4. 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,

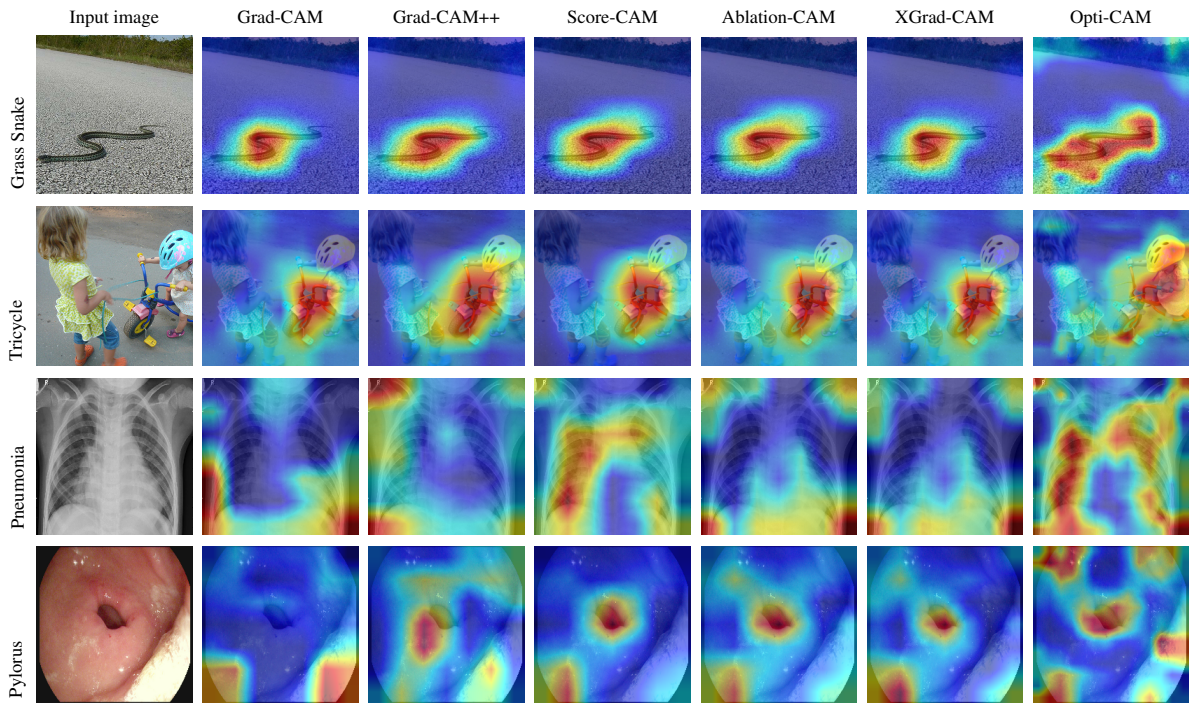


Figure 2: Saliency maps obtained by different methods for ImageNet (top two rows), Chest X-ray (row 3) and Kvasir (row 4) with VGG. Ground truth class shown on the left of the input image.

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 [**? ?** ].

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 **??**, 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

| METHOD | AD↓ | | | AG↑ | | | AI↑ | | |
|---|---|---|---|---|---|---|---|---|---|
| | $S$ | $B\cap S$ | $S\setminus B$ | $S$ | $B\cap S$ | $S\setminus B$ | $S$ | $B\cap S$ | $S\setminus B$ |
| $S := B$ | 67.2 | – | – | 2.3 | – | – | 9.2 | – | – |
| $S := I \setminus B$ | 44.0 | – | – | 2.8 | – | – | 16.3 | – | – |
| Fake-CAM [**?** ] | 0.5 | 67.2 | 44.1 | 0.7 | 2.3 | 2.8 | 42.0 | 9.2 | 18.9 |
| Grad-CAM [**?** ] | 15.0 | 72.6 | 52.1 | 15.3 | 1.8 | 6.0 | 40.4 | 8.4 | 19.4 |
| Grad-CAM++ [**?** ] | 16.5 | 72.9 | 53.1 | 10.6 | 1.6 | 4.1 | 35.2 | 7.3 | 17.1 |
| Score-CAM [**?** ] | 12.5 | 71.5 | 50.5 | 16.1 | 2.2 | 6.3 | 42.5 | 8.6 | 20.8 |
| Ablation-CAM [**?** ] | 15.1 | 72.8 | 52.1 | 13.5 | 1.7 | 5.6 | 39.9 | 7.8 | 19.0 |
| XGrad-CAM [**?** ] | 14.3 | 72.6 | 51.4 | 15.1 | 1.8 | 6.0 | 42.1 | 8.0 | 20.1 |
| Layer-CAM [**?** ] | 49.2 | 84.2 | 74.4 | 2.7 | 0.4 | 1.2 | 12.7 | 4.4 | 7.3 |
| ExPerturbation [**?** ] | 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 3: *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 [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better; bold: best, excluding Fake-CAM.

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 (**??**) and normalization (**??**) of the saliency map. More choices of (**??**), layer $\ell$, number of iterations and learning rates, selector function $g_c$ and initialization of $\mathbf{w}$ are studied in the supplementary material.

*Normalization function.* For normalization function $n$ (**??**), we investigate three choices:

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

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

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

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

| METHOD | $F_\ell^c$ | $n$ | AD↓ | AG↑ | AI↑ |
|---|---|---|---|---|---|
| Fake-CAM [**?** ] | | | 0.5 | 0.7 | 42.1 |
| Grad-CAM [**?** ] | | | 15.0 | 15.3 | 40.4 |
| Grad-CAM++ [**?** ] | | | 16.5 | 10.6 | 35.2 |
| Score-CAM [**?** ] | | | 12.5 | 16.1 | 42.6 |
| Ablation-CAM [**?** ] | | | 15.1 | 13.5 | 39.9 |
| XGrad-CAM [**?** ] | | | 14.3 | 15.1 | 42.1 |
| Layer-CAM [**?** ] | | | 49.2 | 2.7 | 12.7 |
| ExPerturbation [**?** ] | | | 43.8 | 7.1 | 18.9 |
| Opti-CAM (ours) | Mask (**??**) | Range (**??**) | **1.4** | **66.3** | **92.5** |
| | Diff (**??**) | Range (**??**) | 7.1 | 18.5 | 54.9 |
| Opti-CAM (ours) | Mask (**??**) | Max (**??**) | 1.6 | 66.2 | 90.3 |
| | Diff (**??**) | Max (**??**) | 6.8 | 17.8 | 54.5 |
| Opti-CAM (ours) | Mask (**??**) | Sigmoid (**??**) | 5.0 | 18.3 | 57.5 |
| | Diff (**??**) | Sigmoid (**??**) | 6.5 | 10.0 | 45.3 |

Table 4: *Ablation study* using VGG16 on 1000 images of ImageNet validation set. AD/AI: average drop/increase [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better; bold: best, excluding Fake-CAM.

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

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

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

*Results.* **??** 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 (**??**). By contrast, the choice of loss function has more impact on performance and we observe that Mask (**??**) 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.

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 Universite - A*Midex, a French Investissements dAvenir 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 **??**. We provide results on more classification metrics in **??**. In **??**, we define localization metrics and provide corresponding results. We provide results on medical data in **??**. We then provide more ablation results in **??**, sanity check in **??**, and results without input image normalization in **??**.

## Appendix A. Implementation details

All input images are resized to $224 \times 224 \times 3$. To optimize the saliency map with Opti-CAM (**??**), we use the Adam [**?** ] 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 (**??**) 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 $\ell$ and the convergence of Opti-CAM are included in **??**.

## Appendix B. Classification metrics

Classification metrics measure the effect on classification performance of masking (element-wise multiplying) the input image by the saliency map. We have used AD, AG and AI in the main paper. Here we discuss Insertion/Deletion [**?** ], providing results and discussing failure cases for Opti-CAM.

### *Appendix B.1. Insertion/Deletion*

*Definition.* Insertion/Deletion [**?** ] 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 **??** for CNNs and transformers. ExPerturbation [**?** ] 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 [**?** ] only performs best on ResNet50. TIBAV [**?** ], 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 **??** 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 **??** indicates, the background is useful in discriminating 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.

### *Appendix B.2. More metrics*

In this section, we show additional metrics including AOPC [**?** ], Max-Sensitivity [**?** ] and ADCC [**?** ]. We use the code and suggested parameters of package Quantus[8] 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,

[8] https://github.com/understandable-machine-intelligence-lab/Quantus

| Method | ResNet50 | | VGG16 | | ViT-B | | DeiT-B | |
|---|---|---|---|---|---|---|---|---|
| | I↑ | D↓ | I↑ | D↓ | I↑ | D↓ | I↑ | D↓ |
| Fake-CAM [**?** ] | 50.7 | 28.1 | 46.1 | 26.9 | 57.4 | 33.3 | 57.5 | 34.2 |
| Grad-CAM [**?** ] | 66.3 | 14.7 | **64.1** | 11.6 | 62.9 | 19.8 | 61.8 | 17.5 |
| Grad-CAM++ [**?** ] | 66.0 | 14.7 | 62.9 | 12.2 | 56.7 | 29.3 | 60.5 | 21.9 |
| Score-CAM [**?** ] | 65.7 | 16.3 | 62.5 | 12.1 | **66.5** | 15.1 | 60.6 | 24.4 |
| XGrad-CAM [**?** ] | 66.3 | 14.7 | **64.1** | 11.7 | 55.6 | 26.5 | 55.2 | 31.1 |
| Layer-CAM [**?** ] | 67.0 | **14.2** | 58.3 | **6.4** | 62.9 | 14.6 | 61.6 | 21.2 |
| ExPerturbation [**?** ] | **70.7** | 15.0 | 61.1 | 15.0 | 64.4 | 18.4 | 62.1 | 27.0 |
| Ablation-CAM [**?** ] | 65.9 | 14.6 | 63.8 | 11.4 | - | - | - | - |
| RawAtt [**?** ] | - | - | - | - | 62.2 | 17.9 | 56.3 | 29.3 |
| Rollout [**?** ] | - | - | - | - | 64.8 | 15.2 | 56.7 | 32.8 |
| TIBAV [**?** ] | - | - | - | - | 66.1 | **14.1** | **63.7** | **16.3** |
| Opti-CAM (ours) | 62.0 | 19.7 | 59.2 | 11.0 | 60.5 | 22.0 | 59.2 | 22.8 |

Table A5: I/D: insertion/deletion [**?** ] scores on ImageNet validation set; ↓ / ↑: lower / higher is better.

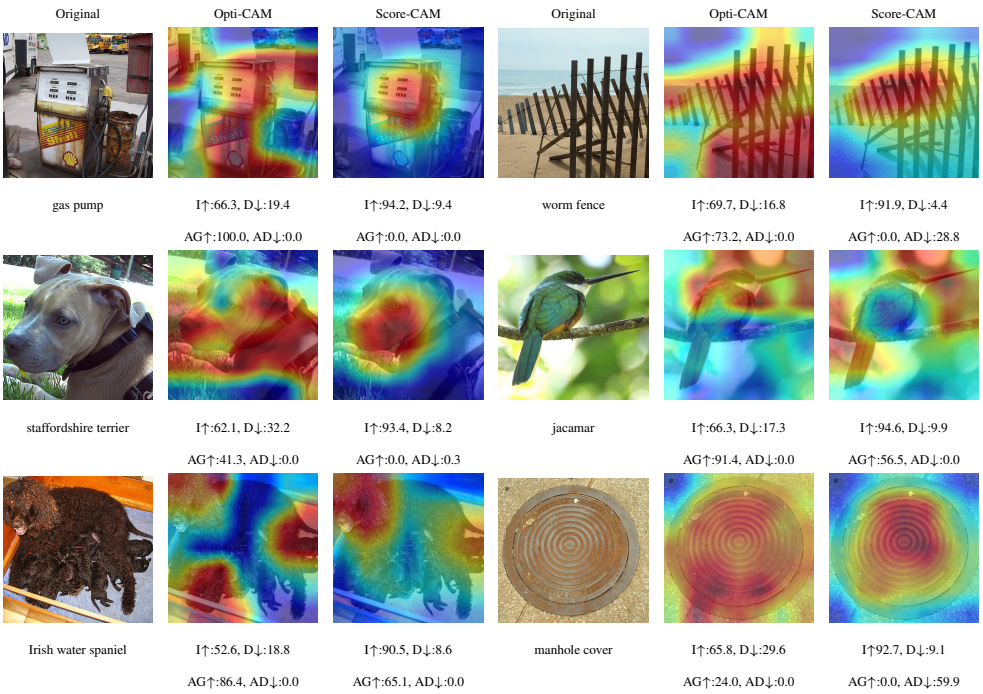


Figure A3: Failure examples of Opti-CAM regarding insertion/deletion.

| METHOD | RESNET50 | | | VGG16 | | |
|---|---|---|---|---|---|---|
| | $AOPC \uparrow$ | $MS \downarrow$ | $ADCC \downarrow$ | $AOPC \uparrow$ | $MS \downarrow$ | $ADCC \downarrow$ |
| Grad-CAM [**?** ] | 11.7 | 1.05 | 74.3 | 13.1 | 1.10 | 73.7 |
| Grad-CAM++ [**?** ] | 11.6 | 1.04 | 73.6 | 11.6 | 1.09 | 74.6 |
| Score-CAM [**?** ] | 10.2 | 1.04 | 61.0 | 11.0 | 1.09 | 73.9 |
| XGrad-CAM [**?** ] | 11.9 | 1.05 | 74.3 | 13.1 | 1.10 | 73.9 |
| Ablation-CAM [**?** ] | 11.1 | 1.04 | 71.5 | 12.5 | 1.10 | 75.5 |
| Layer-CAM [**?** ] | **13.0** | 1.22 | 61.1 | **13.3** | 1.25 | 51.7 |
| ExPerturbation [**?** ] | 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 |

Table A6: *AOPC/MS/ADCC* scores on ImageNet validation set; ↓ / ↑: lower / higher is better.

we use the official code[9]. We evaluate these metrics on ImageNet validation set using ResNet50 and VGG16. The results are reported in **??**. 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.

## Appendix C. 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* $F_1$ *score*, *box accuracy* (BoxAcc) [**?** ], standard pointing game (SP) [**?** ], *energy pointing game* (EP) [**?** ] and *saliency metric* (SM) [**?** ] on the ILSVRC2014[10] 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.

[9] https://github.com/aimagelab/ADCC?fbclid=IwAR0YK_93lxp4pZQnt34SlA9aeNCLRX8m0u8yTZPxbTXi80qiyhTiqxWaQ7o

[10] https://www.image-net.org/challenges/LSVRC/2014/index#

*Appendix C.1. Definitions*

We are given the saliency map $S^c$ obtained from test image $\mathbf{x}$ for ground truth class $c$. We denote by $S^c_{\mathbf{p}}$ 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 (**??**), 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:

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

where IoU is intersection over union.

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

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

*Pixel-wise $F_1$ 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

$$P := \frac{\sum_{\mathbf{p}\in U} S^c_{\mathbf{p}}}{\sum_{\mathbf{p}} S^c_{\mathbf{p}}} \tag{A3}$$

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

$$R := \frac{\sum_{\mathbf{p}\in U} S^c_{\mathbf{p}}}{|U|}. \tag{A4}$$

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

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

After averaging over the test images, we take the maximum of this measure over a set of values $\eta$ and then the average over a set of values $\delta$.

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

$$\mathrm{SP} := \mathbb{1}_{\mathbf{p}^* \in U}. \tag{A6}$$

*Energy Pointing game (EP) [**?** ].* is equivalent to precision (**??**).

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

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

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

| METHOD | RESNET50 | | | | | | | VGG16 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OM↓ | LE↓ | F1↑ | BA↑ | SP↑ | EP↑ | SM↓ | OM↓ | LE↓ | F1↑ | BA↑ | SP↑ | EP↑ | SM↓ |
| Fake-CAM [**?** ] | 63.6 | 54.0 | 57.7 | 47.9 | 99.8 | 28.5 | 0.98 | 64.7 | 54.0 | 57.7 | 47.9 | 99.8 | 28.5 | 1.07 |
| Grad-CAM [**?** ] | 72.9 | 65.8 | 49.8 | **56.2** | 69.8 | 33.3 | 1.30 | 71.1 | 62.3 | 42.0 | 54.2 | 64.8 | 32.0 | 1.39 |
| Grad-CAM++ [**?** ] | 73.1 | 66.1 | **50.4** | **56.2** | 69.9 | 33.1 | 1.29 | 70.8 | 61.9 | 44.3 | 55.2 | 66.2 | 32.3 | 1.38 |
| Score-CAM [**?** ] | **72.2** | 64.9 | 49.6 | 54.5 | 68.7 | 32.4 | **1.25** | 71.2 | 62.5 | **45.3** | **58.5** | **68.2** | 33.4 | 1.40 |
| Ablation-CAM [**?** ] | 72.8 | 65.7 | 50.2 | 56.1 | 69.9 | 33.1 | 1.26 | 71.3 | 62.6 | 43.2 | 56.2 | 65.7 | 32.7 | 1.39 |
| XGrad-CAM [**?** ] | 72.9 | 65.8 | 49.8 | **56.2** | 69.8 | 33.3 | 1.30 | 70.8 | 62.0 | 41.9 | 53.5 | 64.4 | 31.6 | 1.41 |
| Layer-CAM [**?** ] | 73.1 | 66.0 | 50.1 | 55.5 | **70.0** | 33.0 | 1.29 | 70.5 | 61.5 | 28.0 | 54.7 | 65.0 | 32.4 | 1.45 |
| ExPerturbation [**?** ] | 73.6 | 66.6 | 37.5 | 44.2 | 64.8 | **38.2** | 1.59 | 74.1 | 66.4 | 37.8 | 43.3 | 62.7 | **36.1** | 1.74 |
| Opti-CAM (ours) | **72.2** | **64.8** | 47.3 | 49.2 | 59.4 | 30.5 | 1.34 | **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 $F_1$ 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 C.2. Results*

We evaluate the localization ability of saliency maps obtained by our Opti-CAM and we compare with other attribution methods quantitatively. **??** and **??** 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.

| METHOD | ViT-B | | | | | | | DeiT-B | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OM↓ | LE↓ | F1↑ | BA↑ | SP↑ | EP↑ | SM↓ | OM↓ | LE↓ | F1↑ | BA↑ | SP↑ | EP↑ | SM↓ |
| Fake-CAM [**?** ] | 62.8 | 54.0 | 57.7 | 47.9 | 99.8 | 28.6 | 0.87 | 61.4 | 54.0 | 57.7 | 47.9 | 99.8 | 28.7 | 0.83 |
| Grad-CAM [**?** ] | 79.6 | 74.3 | 29.4 | 45.0 | 58.1 | 31.0 | 3.27 | 65.5 | 60.3 | 44.3 | 47.2 | 62.8 | 30.2 | 1.20 |
| Grad-CAM++ [**?** ] | 84.2 | 80.6 | 14.8 | 23.8 | 51.4 | 27.3 | 4.15 | 70.6 | 67.2 | 34.3 | 43.6 | 57.7 | 30.3 | 2.14 |
| Score-CAM [**?** ] | 77.6 | 71.6 | 46.0 | 54.3 | **66.1** | 33.1 | 3.14 | 79.9 | 76.2 | 31.9 | 43.8 | **63.4** | 32.2 | 3.14 |
| XGrad-CAM [**?** ] | 82.0 | 76.9 | 19.6 | 41.3 | 52.8 | 28.5 | 3.31 | 82.0 | 78.4 | 19.5 | 44.1 | 53.4 | 28.8 | 3.03 |
| Layer-CAM [**?** ] | 70.7 | 63.9 | 20.6 | 50.5 | 60.7 | 32.6 | 1.44 | 80.2 | 77.3 | 17.6 | 50.8 | 62.7 | 35.1 | 3.15 |
| ExPerturbation [**?** ] | 71.5 | 64.9 | 35.9 | 44.6 | 62.3 | **35.3** | 1.34 | 69.9 | 64.3 | 36.2 | 44.2 | 63.1 | **35.5** | 1.16 |
| RawAtt [**?** ] | 72.4 | 64.8 | 18.5 | 50.4 | 55.4 | 31.6 | 1.68 | 73.5 | 68.2 | 5.9 | **48.1** | 46.5 | 27.3 | 1.91 |
| Rollout [**?** ] | 67.6 | 58.8 | 36.9 | 50.7 | 57.8 | 30.0 | 1.16 | 63.9 | 57.0 | 27.8 | 47.9 | 36.5 | 27.2 | 0.94 |
| TIBAV [**?** ] | 70.1 | 63.1 | 26.6 | **58.8** | **66.1** | 35.0 | 1.23 | 68.2 | 62.2 | 28.1 | 59.6 | 64.1 | 33.5 | 1.08 |
| Opti-CAM (ours) | **64.4** | **54.6** | **54.5** | 48.0 | 58.2 | 28.7 | **0.98** | **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* $F_1$ *score*; BA: box accuracy; SP: standard pointing game; EP: energy pointing game; SM: *saliency metric*. ↓ / ↑: lower / higher is better. Bold: best, excluding Fake-CAM.

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 D. 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 D.1. Datasets*

*Chest X-ray.* [**?** ] 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.

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

Table A9: *Classification metrics* on Chest X-ray and KVASIR datasets. AD/AI: average drop/increase [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better; Bold: best, excluding Fake-CAM.

*Kvasir.* [**?** ] 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 D.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 $5 \times 10^{-3}$ 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 D.3. Results*

**??** reports metrics AD/AG/AI and **??** 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

| Method | Chest X-ray | | | | Kvasir | | | |
|---|---|---|---|---|---|---|---|---|
| | ResNet50 | | VGG16 | | ResNet50 | | VGG16 | |
| | I↑ | D↓ | I↑ | D↓ | I↑ | D↓ | I↑ | D↓ |
| Grad-CAM [**?** ] | 83.0 | 75.7 | 85.0 | 81.9 | **81.3** | **32.2** | 72.1 | 48.9 |
| Grad-CAM++ [**?** ] | 82.2 | 79.1 | 85.1 | 81.8 | 80.2 | 33.8 | 72.1 | 48.7 |
| Score-CAM [**?** ] | 82.9 | 77.0 | 87.6 | 79.0 | 80.6 | 33.4 | 79.3 | 34.9 |
| Ablation-CAM [**?** ] | **83.5** | **75.1** | **92.0** | **73.1** | 80.3 | 32.6 | **79.4** | 36.2 |
| XGrad-CAM [**?** ] | 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 A10: I/D: insertion/deletion [**?** ] on Chest X-ray and KVASIR dataset using both ResNet50 and VGG16. ↓ / ↑: lower / higher is better.

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.

## Appendix E. More ablations

### *Appendix E.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 $\alpha$ that takes values in $\{0.01, 0.05, 0.1, 0.5, 1, 1.5, 2, 3, 5, 10\}$. When $\alpha$ is small, the saliency maps become more uniform, so that more information about the original image is revealed to the network. Respectively, when $\alpha$ is large, the saliency maps become more selective, so that the network sees fewer parts of the input. The order of pixels is maintained.

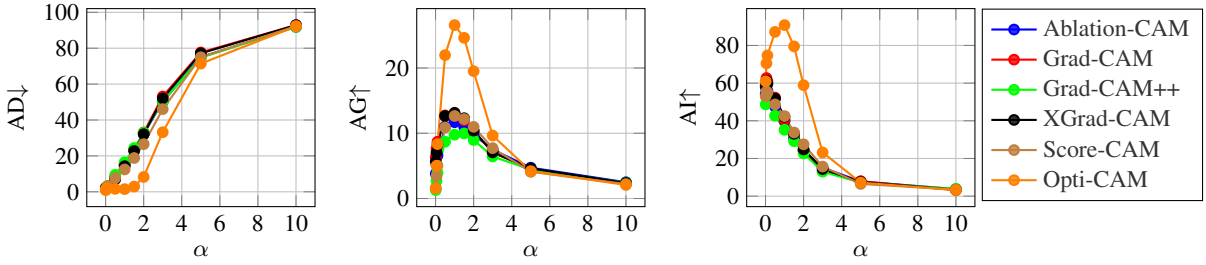


Figure A4: Effect of *selectivity* (raising element-wise to exponent $\alpha$) of saliency maps on classification performance. AD/AI: average drop/increase [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better.

Results in terms of AD, AG, AI are shown in **??**, averaged over $1,000$ ImageNet images. We observe that AD stays near zero for Opti-CAM for $\alpha < 2$, while it increases linearly with $\alpha$ 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 E.2. Opti-CAM components*

*Objective function.* We consider more alternative definitions of the objective function $F_\ell^c$, 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

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

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

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

According to **??**, 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.* **??** shows how the performance of Opti-CAM, in terms of AD/AI/AG, depends on the layer $\ell$ of the VGG16 network used to compute the saliency map $S_\ell^c$ (**??**). 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 [**? ? ? ?** ].

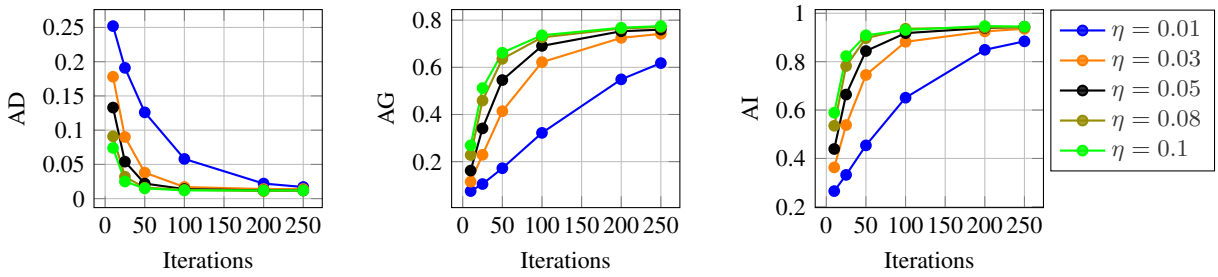


Figure A5: Classification metrics *vs.* number of iterations for different learning rates, using VGG-16 on 1000 images of ImageNet. AD/AI: average drop/increase [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better.

| METHOD | $F_\ell^c$ | AD↓ | AG↑ | AI↑ |
|---|---|---|---|---|
| Fake-CAM [**?** ] | | 0.5 | 0.7 | 42.1 |
| Grad-CAM [**?** ] | | 15.0 | 15.3 | 40.4 |
| Grad-CAM++ [**?** ] | | 16.5 | 10.6 | 35.2 |
| Score-CAM [**?** ] | | 12.5 | 16.1 | 42.6 |
| Ablation-CAM [**?** ] | | 15.1 | 13.5 | 39.9 |
| XGrad-CAM [**?** ] | | 14.3 | 15.1 | 42.1 |
| Layer-CAM [**?** ] | | 49.2 | 2.7 | 12.7 |
| ExPerturbation [**?** ] | | 43.8 | 7.1 | 18.9 |
| Opti-CAM | Mask (**??**) | 1.4 | **66.3** | 92.5 |
| | Diff (**??**) | 7.1 | 18.5 | 54.9 |
| | IOMask (**??**) | **0.2** | 5.5 | **99.7** |
| | IODiff (**??**) | 25.9 | 7.6 | 42.6 |

Table A11: *Ablation study on objective function* using VGG16 on 1000 images of ImageNet validation set. Choices for objective function $F_\ell^c$: Mask: (**??**); Diff: (**??**); IOMask: (**??**); IODiff: (**??**). Choice for normalization function $n$: Range (**??**). Iterations: 50. AD/AI: average drop/increase [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better.

| LAYER | AD↓ | AG↑ | AI↑ | LAYER | AD↓ | AG↑ | AI↑ |
|---|---|---|---|---|---|---|---|
| 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 A12: *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 [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better.

*Convergence.* Finally, **??** 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 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.

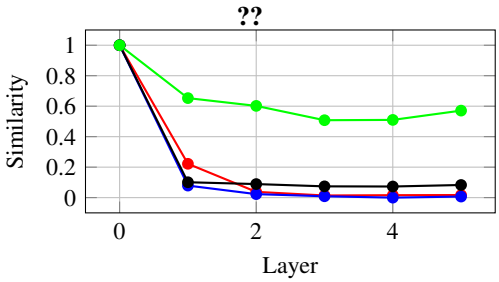


Figure A6: *Sanity check* of Opti-CAM on $1,000$ images of ImageNet validation set using ResNet50. Similarity between saliency maps by original and randomized network, where layers are progressively replaced by random ones.

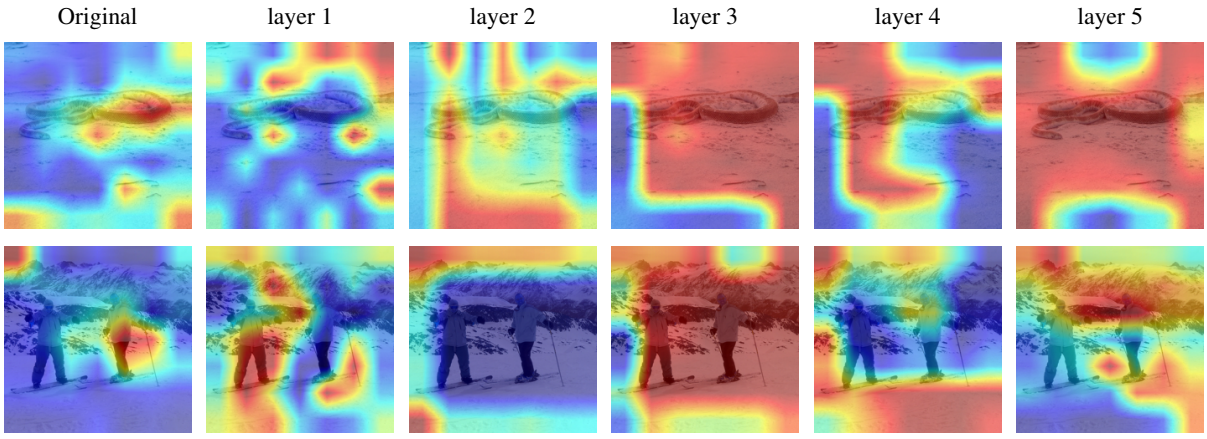


Figure A7: *Sanity check visualization* of Opti-CAM on two images of ImageNet validation set using ResNet50. First column: Opti-CAM saliency maps for the original network; remaining columns: Opti-CAM saliency maps where layers are progressively replaced by random ones.

## Appendix F. Sanity check

We use the model parameter randomization test proposed by [**?** ]. 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 [**?** ], 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 **??** (saliency map similarity measurements) and **??** (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 models[11]. The SSIM similarity is $0.013$, which shows that Opti-CAM is sensitive to the relationship between instances and labels.

| METHOD | RESNET50 | | | | VGG16 | | | |
|---|---|---|---|---|---|---|---|---|
| | AD↓ | AG↑ | AI↑ | T | AD↓ | AG↑ | AI↑ | T |
| Fake-CAM [**?** ] | 0.9 | 0.7 | 47.4 | 0.00 | 0.5 | 0.3 | 47.7 | 0.00 |
| Grad-CAM [**?** ] | 36.4 | 5.5 | 27.0 | 0.03 | 41.6 | 3.3 | 25.2 | 0.02 |
| Grad-CAM++ [**?** ] | 37.6 | 4.9 | 24.0 | 0.04 | 46.3 | 2.0 | 19.0 | 0.02 |
| Score-CAM [**?** ] | 28.8 | 8.8 | 33.6 | 20.47 | 39.3 | 3.5 | 24.6 | 3.08 |
| Ablation-CAM [**?** ] | 36.6 | 5.1 | 25.6 | 18.49 | 41.8 | 2.9 | 24.0 | 2.95 |
| XGrad-CAM [**?** ] | 36.4 | 5.5 | 27.0 | 0.03 | 40.6 | 3.4 | 25.8 | 0.02 |
| Layer-CAM [**?** ] | 42.6 | 4.2 | 19.2 | 0.02 | 82.1 | 0.3 | 6.9 | 0.01 |
| ExPerturbation [**?** ] | 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 A13: *Classification metrics* on ImageNet validation set, without input normalization. AD/AI: average drop/increase [**?** ]; AG: average gain (ours); ↓ / ↑: lower / higher is better. T: Average time (sec) per batch of 8 images. Bold: best, excluding Fake-CAM.

[11] https://github.com/pytorch/vision/tree/main/references/classification

## Appendix G. 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 **??** that correspond to **??**. Finally, code is provided to allow for the reproduction and verification of our results.