A. Method Details

Algorithm 1 Training procedure to compute $\mathcal{S}^{=}$ and $\mathcal{S}^{\neq}$

Require: $k \in \mathbb{N}_{\geq 2}$, $\mathcal{D} = \{\mathcal{V}_i\}_{i=1}^{k}$
  $\mathcal{S}^{=} \leftarrow \{\}, \mathcal{S}^{\neq} \leftarrow \{\}$
  for all $i \in \{1, \ \ldots, \ k\}$ **do**
    Train $\boldsymbol{f}_i$ on $\mathcal{D} \setminus \mathcal{V}_i$
    for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$ **do**
      // generate explanations on all dataset
      $\boldsymbol{\phi}_{\boldsymbol{x}}^{(i)} \leftarrow \boldsymbol{\Phi}(\boldsymbol{f}_i, \boldsymbol{x})$
    end for
  end for
  for all $i \in \{1, \ \ldots, \ k\}$ **do**
    for all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{V}_i$ **do**
      for all $j \in \{1, \ \ldots, \ k \mid i \neq j\}$ **do**
        // $\boldsymbol{f}_j$ was trained on $\boldsymbol{x}$, $\boldsymbol{f}_i$ was not
        $\delta_{\boldsymbol{x}}^{(i,j)} \leftarrow d(\boldsymbol{\phi}_{\boldsymbol{x}}^{(i)}, \boldsymbol{\phi}_{\boldsymbol{x}}^{(j)})$
        if $\boldsymbol{f}_i(\boldsymbol{x}) = \boldsymbol{y}$ **and** $\boldsymbol{f}_j(\boldsymbol{x}) = \boldsymbol{y}$ **then**
          // both model are correct
          $\mathcal{S}^{=} \leftarrow \mathcal{S}^{=} \cup \{\delta_{\boldsymbol{x}}^{(i,j)}\}$
        else if $\boldsymbol{f}_i(\boldsymbol{x}) = \boldsymbol{y}$ **or** $\boldsymbol{f}_j(\boldsymbol{x}) = \boldsymbol{y}$ **then**
          // only one model is correct
          $\mathcal{S}^{\neq} \leftarrow \mathcal{S}^{\neq} \cup \{\delta_{\boldsymbol{x}}^{(i,j)}\}$
        end if
      end for
    end for
  end for
  Return $\mathcal{S}^{=}, \mathcal{S}^{\neq}$

B. Explanation methods

In the following section, the formulation of the different methods used is given. As a reminder, we focus on a classification model $\boldsymbol{f} : \mathbb{R}^d \rightarrow \mathbb{R}^C$ where C is the number of classes. We assume $\boldsymbol{f}_c(\boldsymbol{x})$ the logit score (before softmax) for class c. An explanation method provides an attribution $\boldsymbol{\phi} \in \mathbb{R}^d$ for each input feature from a model and an input of interest. Each value then corresponds to the importance of this feature for the model results.

Saliency Map (SA) is a visualization techniques based on the gradient of a class score relative to the input, indicating in an infinitesimal neighborhood, which pixels must be modified to most affect the score of the class of interest.

$$\boldsymbol{\Phi}^{SA}(\boldsymbol{x}) = \left| \frac{\partial \boldsymbol{f}_c(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|$$

Gradient $\odot$ Input (GI) is based on the gradient of a class score relative to the input, element-wise with the input, it was introduced to improve the sharpness of the attribution maps. A theoretical analysis conducted by [3] showed that Gradient $\odot$ Input is equivalent to ϵ-LRP and DeepLIFT methods under certain conditions: using a baseline of zero, and with all biases to zero.

$$\boldsymbol{\Phi}^{GI}(\boldsymbol{x}) = \boldsymbol{x} \odot \left| \frac{\partial \boldsymbol{f}_c(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|$$

Integrated Gradients (IG) consists of summing the gradient values along the path from a baseline state to the current value. The baseline is defined by the user and often chosen to be zero. This integral can be approximated with a set of m points at regular intervals between the baseline and the point of interest. In order to approximate from a finite number of steps, we use a Trapezoidal rule and not a left-Riemann summation, which allows for more accurate results and improved performance (see [50] for a comparison). The final result depends on both the choice of the baseline $\boldsymbol{x}_0$ and the number of points to estimate the integral. In the context of these experiments, we use zero as the baseline and $m = 60$.

$$\boldsymbol{\Phi}^{IG}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_0) \int_0^1 \frac{\partial \boldsymbol{f}_c(\boldsymbol{x}_0 + \alpha(\boldsymbol{x} - \boldsymbol{x}_0))}{\partial \boldsymbol{x}} d\alpha$$

SmoothGrad (SG) is also a gradient-based explanation method, which, as the name suggests, averages the gradient at several points corresponding to small perturbations (drawn i.i.d from a normal distribution of standard deviation σ) around the point of interest. The smoothing effect induced by the average help reducing the visual noise, and hence improve the explanations. In practice, Smoothgrad is obtained by averaging after sampling m points. In the context of these experiments, we took $m = 60$ and $\sigma = 0.2$ as suggested in the original paper.

$$\boldsymbol{\Phi}^{SG}(\boldsymbol{x}) = \mathop{\mathbb{E}}_{\varepsilon \sim \mathcal{N}(0, \boldsymbol{I}\sigma^2)} \left[\frac{\partial \boldsymbol{f}_c(\boldsymbol{x} + \varepsilon)}{\partial \boldsymbol{x}} \right]$$

Grad-CAM (GC) can be used on Convolutional Neural Network (CNN), it uses the gradient and the feature maps $\boldsymbol{A}^{(k)}$ of the last convolution layer. More precisely, to obtain the localization map for a class, we need to compute the weights $\alpha_c^{(k)}$ associated to each of the feature map activation $\boldsymbol{A}^{(k)}$, with k the number of filters and Z the number of features in each feature map we define $\alpha_c^{(k)} = \frac{1}{Z} \sum_i \sum_j \frac{\partial \boldsymbol{f}_c(\boldsymbol{x})}{\partial A_{ij}^{(k)}}$ and

$$\boldsymbol{\Phi}^{GC} = max(0, \sum_k \alpha_c^{(k)} \boldsymbol{A}^{(k)})$$

Notice that the size of the explanation depends on the size (height, width) of the last feature map, a bilinear interpolation is performed in order to find the same dimensions as the input.

Grad-CAM++ (G+) is an extension of Grad-CAM combining the positive partial derivatives of feature maps of a convolutional layer with a weighted special class score. The

weights $\alpha_c^{(k)}$ associated to each feature map is computed as follow :

$$\alpha_k^c = \sum_i \sum_j [\frac{\frac{\partial^2 \boldsymbol{f}_c(\boldsymbol{x})}{(\partial A_{ij}^{(k)})^2}}{2\frac{\partial^2 \boldsymbol{f}_c(\boldsymbol{x})}{(\partial A_{ij}^{(k)})^2} + \sum_i \sum_j A_{ij}^{(k)} \frac{\partial^3 \boldsymbol{f}_c(\boldsymbol{x})}{(\partial A_{ij}^{(k)})^3}}]$$

RISE (RI) is a black-box method that consist of probing the model with randomly masked versions of the input image to deduce the importance of each pixel using the corresponding outputs. The binary masks $\boldsymbol{m} \sim \mathcal{M}$ are generated in a subspace of the input space, then upsampled with a bilinear interpolation (once upsampled the masks are no longer binary).

For ImageNet the number of masks was $m = 4000$, for all the other datasets $m = 1000$.

$$\boldsymbol{\Phi}^{RI}(\boldsymbol{x}) = \frac{1}{\mathbb{E}(\mathcal{M})N} \sum_{i=0}^{N} \boldsymbol{f}_c(\boldsymbol{x} \odot \boldsymbol{m_i})\boldsymbol{m_i}$$

C. Fidelity

Various fidelity metrics have been proposed that essentially measure the correlation between input variables and the drop in score when these variables are set to a baseline state [40, 60, 38, 35]. In this work, we use μF from [6]:

$$\mu F = \underset{\substack{S \subseteq \{1,\dots,d\} \\ |S|=k}}{\text{Corr}} \left(\sum_{i \in S} \boldsymbol{\Phi}(\boldsymbol{f}, \boldsymbol{x})_i, \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}_{[\boldsymbol{x}_i = \bar{\boldsymbol{x}}_i, i \in S]}) \right) \tag{7}$$

Where $\boldsymbol{f}$ is a predictor, $\boldsymbol{\Phi}$ an explanation function, S a subset indices of $\boldsymbol{x}$ and $\bar{\boldsymbol{x}}$ a baseline reference. The choice of a proper baseline is still an active area of research [52].

D. Considered measures for ReCo

As mentioned in when introducing ReCo, one would be tempted to use directly a distance between distributions, we briefly explain why we did not make this choice. In addition, we detail an alternative measure, also based on balanced accuracy, which gives consistent results.

A first intuition to measure the shift between the $\boldsymbol{\mathcal{S}}^{=}$ and $\boldsymbol{\mathcal{S}}^{\neq}$ histograms would be to consider the usual measures, such as Kullback-Leibler (KL) divergence.

However, these distances are problematic in that the order of the distributions actually matters more than the distance between them, and these two measures can give a good score even when the explanations are inconsistent Similarly, considering the 1-Wasserstein measure, we could construct an inconsistent case by exploiting the invariance to the direction of transport. For these reasons, we have therefore chosen a classification measure, based on maximizing balanced accuracy. Nevertheless, one could also (observing similar results) use the area under the curve (AUC) of the balanced accuracy, such as :

$$ReCo_{AUC} = \frac{1}{|\boldsymbol{\mathcal{S}}|} \sum_{\gamma \in \boldsymbol{\mathcal{S}}} TPR(\gamma) + TNR(\gamma) - 1$$

Table 4. 1-Lipschitz model architecture for Cifar10.

Conv2D(48)
PReLU
AvgPooling2D((2, 2))
Dropout(0.2)
Conv2D(96)
PReLU
AvgPooling2D((2, 2))
Dropout(0.2)
Conv2D(96)
AvgPooling2D((2, 2))
Flatten
Dense(10)

E. Models

As mentioned in the paper, the models used are all (with the exception of 1-Lipschitz networks) ResNet-18, with variations in size and number of filters used. Preserving the increase of filters at each depth by the original factor (x2), we took care to define for each dataset, a base filters value, as the number of filters for the first convolution layer. Another difference concerns the dropout rates used, indeed we had dropout to improve the performance of the tested models. Moreover, it should be remembered that there is no difference in architecture between the normally trained models and the degraded models.

We report here the architecture of the models for each of the datasets:
Fashion-MNIST base filters 26, Dropout 0.4 (92%, $\pm$1%)
EuroSAT base filters 46, Dropout 0.25 (95%, $\pm$1%)
Cifar10 base filters 32, Dropout 0.25 (78%, $\pm$4%)
ImageNet ResNet50 (88%, $\pm$3%)

E.1. Lipschitz models

The 1-Lipschitz models use spectral regularization on the Dense and Convolutions layers. The architecture is as described in Table 4.

E.2. Randomization test

For the randomisation of the model weights, we added noise drawn from a normal distribution $\varepsilon \sim \mathcal{N}(0, 0.5)$ to each convolution layer, with the intensity of the degradation impacting on the number of parameters affected by this noise.

F. Distances tests

F.1. Spatial correlation

The first test concerns the spatial distance between two areas of interest for an explanation. It is desired that the spatial distance between areas of interest be expressed by the distance used. As a results, two different but spatially close explanations should have a low distance. The test consists in generating several masks representing a point of interest, starting from a left corner of an image of size (32 x 32) and moving towards the right corner by interpolating 100 different masks. The distance between the first image and each interpolation is then measured (see Fig. 6).

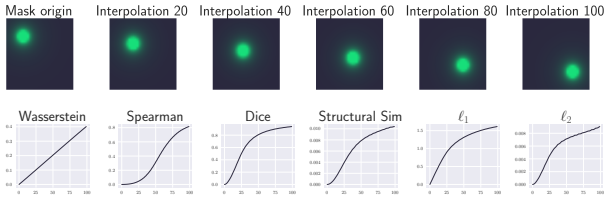

Figure 6. Distances with moving interest point. The first line shows the successive interpolations between the baseline image (left), and the target image (right). The second line shows the evolution of the distance between each interpolation and the baseline image.

The different distances evaluated pass this sanity check, i.e. a monotonous growth of the distance, image of the spatial distance of the two points of interest.

F.2. Noise test

The second test concerns the progressive addition of noise. It is desired that the progressive addition of noise to an original image will affect the distance between the original noise-free image and the noisy image. Formally, with x the original image, and $\varepsilon \sim \mathcal{N}(0, \boldsymbol{I}\sigma^2)$ an isotropic Gaussian noise, we wish the distance d to show a monotonic positive correlation $\text{corr}(dist(x, x+\varepsilon), \varepsilon)$.

In order to validate this prerogative, a Gaussian noise with a progressive intensity σ is added to an original image, and the distance between each of the noisy images and the original image is measured. For each value of σ the operation is repeated 50 times.

Over the different distances tested, they all pass the sanity test : there is a monotonous positive correlation (as seen in Fig. 7). Although SSIM and ℓ_2 have a higher variance.

One will nevertheless note the instability of the Dice score in cases where the areas of interest have a low surface area, as well as a significant computation cost for the Wasserstein distance. For all these reasons, we chose to stay in line with previous work using the absolute value of Spearman rank correlation.

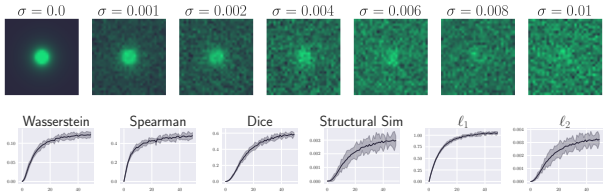

Figure 7. Distances with noisy images. The first line shows original noise-free image (left) and noisy copies computed by increasing σ. The second line shows the distances between each noisy image and the baseline image.

G. Additional results

Metrics	IG	SG	SA	GI	GC	G+	RI
μF	0.11	0.31	0.23	0.10	**0.91**	<u>0.89</u>	0.84
MeGe	0.58	0.46	0.45	0.55	<u>0.72</u>	**0.82**	0.56
ReCo	0.11	0.15	0.15	0.09	**0.64**	0.49	<u>0.52</u>

Table 5. *Fidelity*, *Consistency* and *Generalizability* score for ResNet-18 models on Cifar10. Higher is better. The first and second best results are respectively in **bold** and <u>underlined</u>.

Metrics	IG	SG	SA	GI	GC	G+	RI
MeGe	0.40	<u>0.42</u>	0.41	0.41	**0.67**	**0.67**	0.39
ReCo	0.31	0.18	0.18	0.23	<u>0.59</u>	**0.64**	0.34

Table 6. *Consistency* and *Generalizability* score for ResNet-18 models on Eurosat. Higher is better. The first and second best results are respectively in **bold** and <u>underlined</u>.

Metrics	IG	SG	SA	GI	GC	G+	RI
MeGe	**0.90**	0.36	0.30	**0.90**	0.77	<u>0.84</u>	0.52
ReCo	<u>0.37</u>	0.13	0.10	<u>0.37</u>	**0.52**	0.32	<u>0.37</u>

Table 7. *Consistency* and *Generalizability* score for ResNet-18 models on Fashion-MNIST. Higher is better. The first and second best results are respectively in **bold** and <u>underlined</u>.

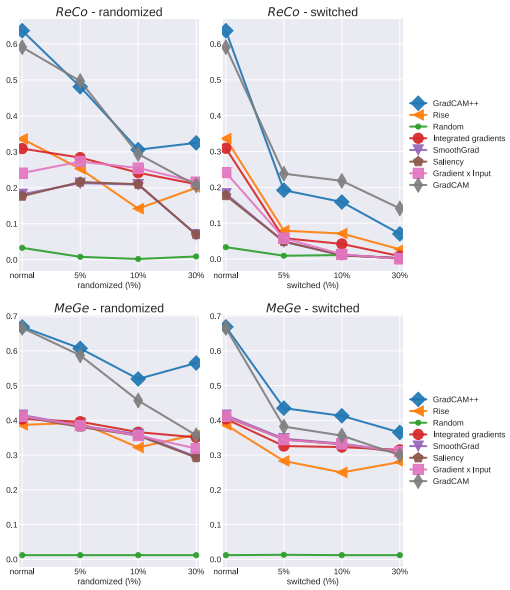

Figure 8. **Eurosat** MeGe and ReCo scores for normally trained models (first point from the left), as well as for progressively randomized models and models trained with switched labels.

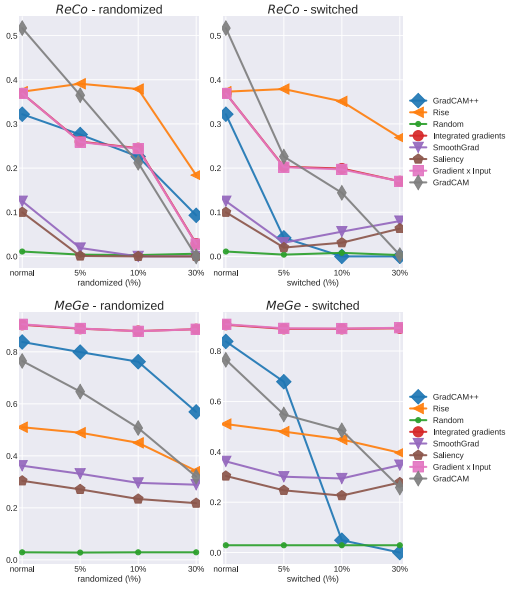

Figure 9. **Fashion-MNIST** MeGe and ReCo scores for normally trained models (first point from the left), as well as for progressively randomized models and models trained with switched labels.

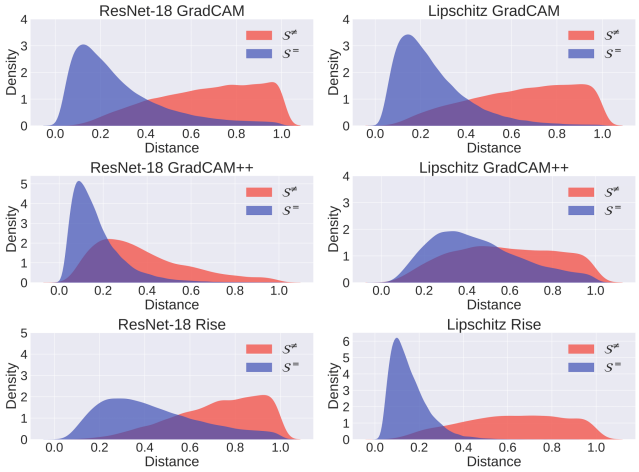

Figure 10. $\mathcal{S}^{=}$ and $\mathcal{S}^{\neq}$ for ResNet (left column) and 1-Lipschitz models (right column) on Cifar10. As explained in this paper, a clear separation between the $\mathcal{S}^{=}$ and $\mathcal{S}^{\neq}$ histograms is a sign of consistent explanations.