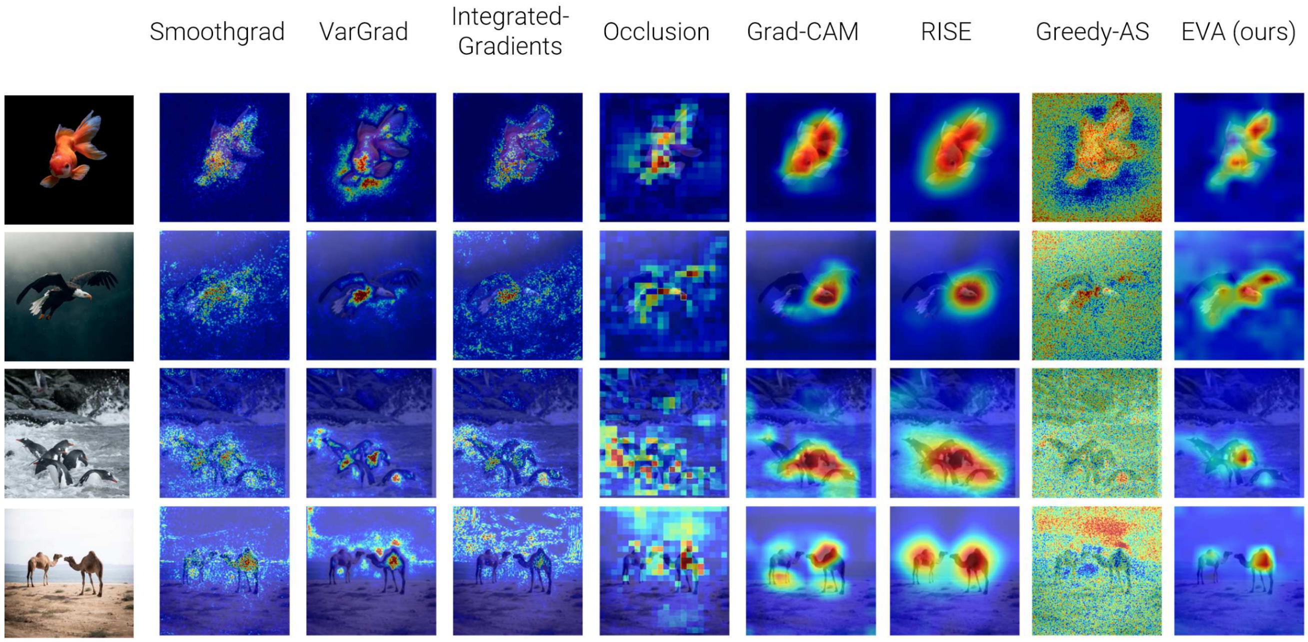

Figure 5. **Qualitative comparison** with other attribution methods. To allow for better visualization, the gradient-based methods (Saliency, Gradient-Input, SmoothGrad, Integrated-Gradient, VarGrad) are clipped at the 2nd percentile. For more results and details on each method and choice of hyperparameters, see Appendix.

	MNIST					Cifar-10					ImageNet				
	Del.↓	Ins.↑	Fid.↑	Rob.↓	Time	Del.↓	Ins.↑	Fid.↑	Rob.↓	Time	Del.↓	Ins.↑	Fid.↑	Rob.↓	Time
Greedy-AS [29]	.260	.497	.110	**.061**	335	.205	.264	-.003	**.013**	4618	.088	.047	.023	**.612**	180056
Greedy-AO	.237	.572	.244	.063	290	**.162**	.283	.041	.024	2874	.086	.050	.023	.752	26762
EVA$_{emp}$	.101	.621	.378	.067	14.4	.184	.270	**.397**	.022	186.6	.070	**.289**	.048	.758	6454
EVA (ours)	**.089**	**.736**	**.428**	.069	1.29	**.164**	.290	**.352**	.025	12.7					

Table 3. Results on Deletion (Del.), Insertion (Ins.), μFidelity (Fid.) and Robustness-S$_r$ (Rob.) metrics. Time in seconds corresponds to the generation of 100 explanations on an Nvidia P100. Note that EVA is the only method with guarantees that the entire set of possible perturbations has been exhaustively searched. Verified perturbation analysis with IBP + Forward + Backward is used for MNIST, with Forward only for CIFAR-10 and with our hybrid strategy described in Section.3.4 for ImageNet. Grad-CAM and Grad-CAM++ are not calculated on the MNIST dataset since the network only has dense layers. Greedy-AO is the equivalent of Greedy-AS but with the *AO* estimator. The first and second best results are in **bold** and underlined, respectively.

A. Qualitative comparison

Regarding the visual consistency of our method, Figure 5 shows a side-by-side comparison between our method and the attribution methods tested in our benchmark. To allow better visualization, the gradient-based methods were 2 percentile clipped.

B. Ablation studies

For a more thorough understanding of the impact of the different components that made EVA - the adversarial overlap and the use of verification tools- we proposed different ablation versions of EVA which are the following: (***i***) Empirical EVA, (***ii***) GreedyAO which is the equivalent of Greedy-AS but with the *AO* estimator. This allow us to perform ablation on the proposed *AO* estimator. Results can be found in Table 3.

B.1. Empirical EVA.

In this section, we describe the ablation consisting in estimating EVA without any use of verified perturbation analysis – thus without any guarantees.

A first intuitive approach would be to replace verification perturbation analysis with adversarial attacks (as used in *Greedy-AS* [29]); we denote this approach as *Greedy-AO*. In addition, we go further with a purely statistical approach based on a uniform sampling of the domain; we denote this approach EVA$_{\text{emp}}$.

This estimator proves to be a very good alternative in terms of computation time but also with respect to the considered metrics as shown in Section 4. Unfortunately the lack of guarantee makes it not as relevant as EVA. Formally, it consists in directly estimating empirically AO using N randomly sampled perturbations.

$$\hat{AO}(\boldsymbol{x}, \mathcal{B}) = \max_{\substack{\boldsymbol{\delta}_1, \cdots \boldsymbol{\delta}_i, \cdots \boldsymbol{\delta}_N \overset{\text{iid}}{\sim} U(\mathcal{B}) \\ c' \neq c}} \boldsymbol{f}_{c'}(\boldsymbol{x} + \boldsymbol{\delta}_i) - \boldsymbol{f}_c(\boldsymbol{x} + \boldsymbol{\delta}_i). \tag{3}$$

We then denote accordingly EVA$_{\text{emp}}$ which uses $\hat{AO}$:

$$\text{EVA}_{\text{emp}}(\boldsymbol{x}, \boldsymbol{u}, \mathcal{B}) = \hat{AO}(\boldsymbol{x}, \mathcal{B}) - \hat{AO}(\boldsymbol{x}, \mathcal{B}_{\boldsymbol{u}}) \tag{4}$$

C. EVA and $Robustness$-$S_{\mathbf{r}}$

We show here that the explanations generated by EVA provide an optimal solution from a certain stage to the $Robustness$-S_{r} metric proposed by [29]. We admit a unique closest adversarial perturbation $\boldsymbol{\delta}^* = \min ||\boldsymbol{\delta}||_p : \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{\delta}) \neq \boldsymbol{f}(\boldsymbol{x})$, and we define ε, the radius of $\mathcal{B}$ as $\varepsilon = ||\boldsymbol{\delta}||_p$. Note that $||\boldsymbol{\delta}||_p$ can be obtained by binary search using the verified perturbation analysis method.

We briefly recall the $Robustness$-S_{r} metric. With $\boldsymbol{x} = (x_1, ..., x_d)$, the set $\mathcal{U} = \{1, ..., d\}$, $\boldsymbol{u}$ a subset of $\mathcal{U}$: $\boldsymbol{u} \subseteq \mathcal{U}$ and $\overline{\boldsymbol{u}}$ its complementary. Moreover, we denote the minimum distance to an adversarial example $\varepsilon^*_{\boldsymbol{u}}$:

$$\varepsilon^*_{\boldsymbol{u}} = \left\{ \min ||\boldsymbol{\delta}||_p \ : \ \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{\delta}) \neq \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\delta}_{\overline{\boldsymbol{u}}} = 0 \right\}$$

The $Robustness$-S_{r} score is the AUC of the curve formed by the points $\{(1, \varepsilon^*_{(1)}), ..., (d, \varepsilon^*_{(d)})\}$ where $\varepsilon^*_{(k)}$ is the minimum distance to an adversarial example for the k most important variables. From this, we can deduce that $||\boldsymbol{\delta}^*|| \leq \varepsilon^*_{\boldsymbol{u}}, \forall \boldsymbol{u} \subseteq \{1, ..., d\}$.

The goal here is to minimize this score, which means for a number of variables $|\boldsymbol{u}| = k$, finding the set of variables $\boldsymbol{u}^*$ such that $\varepsilon^*_{\boldsymbol{u}}$ is minimal. We call this set the *optimal set at k*.

Definition C.1. The *optimal set at k* is the set of variables $\boldsymbol{u}^*_k$ such that

$$\boldsymbol{u}^*_k = \underset{\boldsymbol{u} \subseteq \mathcal{U}, \ |\boldsymbol{u}| = k}{\arg\min} \ \varepsilon^*_{\boldsymbol{u}}.$$

We note that finding the minimum cardinal of a variable to guarantee a decision is also a standard research problem [32, 33] and is called subset-minimal explanations.

Intuitively, the optimal set is the combination of variables that allows finding the closest adversarial example. Thus, minimizing $Robustness$-S_{r} means finding the optimal set $\boldsymbol{u}^*$ for each k. Note that this set can vary drastically from one step to another, it is therefore potentially impossible for attribution to satisfy this optimality criterion at each step. Nevertheless, an optimal set that is always reached at some step is the one allowing to build $\boldsymbol{\delta}^*$. We start by defining the notion of an essential variable before showing the optimality of $\boldsymbol{\delta}^*$.

Definition C.2. Given an adversarial perturbation $\boldsymbol{\delta}$, we call *essentials variables* $\boldsymbol{u}$ all variables such that $|\boldsymbol{\delta}_i| > 0, i \in \boldsymbol{u}$. Conversely, we call *inessentials variables* variables that are not essential.

For example, if $\boldsymbol{\delta}^*$ has k *essential variables*, it is reachable by modifying only k variables. This allows us to characterize the optimal set at step k.

Proposition C.3. *Let $\boldsymbol{u}$ be the set of essential variables of $\boldsymbol{\delta}^*$, then $\boldsymbol{u}$ is an optimal set for k, with $k \in [\![|\boldsymbol{u}|, d]\!]$.*

Proof. Let $\boldsymbol{v}$ be a set such that $\varepsilon^*_{\boldsymbol{v}} < \varepsilon^*_{\boldsymbol{u}}$, then $\varepsilon^*_{\boldsymbol{v}} < ||\boldsymbol{\delta}^*||$ which is a contradiction. □

Specifically, as soon as we have the variables allowing us to build $\boldsymbol{\delta}^*$, then we reach the minimum possible for $Robustness$-S_{r}. We will now show that EVA allows us to reach this in $|\boldsymbol{u}|$ steps, with $|\boldsymbol{u}| \leq d$ by showing (1) that $\boldsymbol{\delta}^*$ *essential variables* obtain a positive attribution and (2) that $\boldsymbol{\delta}^*$ *inessential variables* obtain a zero attribution.

Proposition C.4. *All essential variables $\boldsymbol{u}$ w.r.t $\boldsymbol{\delta}^*$ have a strictly positive importance score EVA$(\boldsymbol{u}) > 0$.*

Proof. Let us assume that i is *essential* and EVA$(i) = 0$, then $\boldsymbol{F}(\mathcal{B}) = \boldsymbol{F}(\mathcal{B}_i)$ which implies

$$\max_{\substack{\boldsymbol{\delta} \in \mathcal{B} \\ c' \neq c}} \boldsymbol{f}_{c'}(\boldsymbol{x}+\boldsymbol{\delta}) - \boldsymbol{f}_c(\boldsymbol{x}+\boldsymbol{\delta}) = \max_{\substack{\boldsymbol{\delta}' \in \mathcal{B}_i \\ c' \neq c}} \boldsymbol{f}_{c'}(\boldsymbol{x}+\boldsymbol{\delta}') - \boldsymbol{f}_c(\boldsymbol{x}+\boldsymbol{\delta}')$$

by uniqueness of the adversarial perturbation, $\boldsymbol{\delta} = \boldsymbol{\delta}'$ which is a contradiction as $\boldsymbol{\delta}' \notin \mathcal{B}_i$ since $\boldsymbol{\delta}'_i \neq 0$ by definition of an *essential variable*. Thus x_i cannot be *essential*, which is a contradiction. □

Essentially, if the variable i is necessary to reach $\boldsymbol{\delta}^*$, then removing it prevents the adversarial example from being reached and lowers the *adversarial overlap*, giving a strictly positive attribution.

Proposition C.5. *All inessential variables $\boldsymbol{v}$ w.r.t. $\boldsymbol{\delta}^*$ have a zero importance score EVA$(\boldsymbol{v}) = 0$.*

Proof. With i being an *inessential* variable, then $\boldsymbol{\delta}^*_i = 0$. It follow that $\boldsymbol{\delta}^* \in \mathcal{B}_i \subseteq \mathcal{B}$. Thus

$$\begin{aligned} \boldsymbol{F}(\mathcal{B}) &= \max_{\substack{\boldsymbol{\delta} \in \mathcal{B} \\ c' \neq c}} \boldsymbol{f}_{c'}(\boldsymbol{x} + \boldsymbol{\delta}) - \boldsymbol{f}_c(\boldsymbol{x} + \boldsymbol{\delta}) \\ &= \boldsymbol{f}_{c'}(\boldsymbol{x} + \boldsymbol{\delta}^*) - \boldsymbol{f}_c(\boldsymbol{x} + \boldsymbol{\delta}^*) \end{aligned}$$

as $\boldsymbol{\delta}^*$ is the unique adversarial perturbation in $\mathcal{B}$, similarly

$$\begin{aligned}\boldsymbol{F}(\mathcal{B}_i) &= \max_{\substack{\boldsymbol{\delta}' \in \mathcal{B} \\ c' \neq c}} \boldsymbol{f}_{c'}(\boldsymbol{x} + \boldsymbol{\delta}') - \boldsymbol{f}_c(\boldsymbol{x} + \boldsymbol{\delta}') \\ &= \boldsymbol{f}_{c'}(\boldsymbol{x} + \boldsymbol{\delta}^*) - \boldsymbol{f}_c(\boldsymbol{x} + \boldsymbol{\delta}^*)\end{aligned}$$

thus $\boldsymbol{F}(\mathcal{B}) = \boldsymbol{F}(\mathcal{B}_i)$ and $\text{EVA}(i) = 0$. □

Finally, since EVA ranks the *essential variables* of $\boldsymbol{\delta}^*$ before the *inessential variables*, and since $\boldsymbol{\delta}^*$ is the *optimal set* from the step $|\boldsymbol{u}|$ to the last one d, then EVA provide the *optimal set*, at least from the step $|\boldsymbol{u}|$.

Theorem C.6. ***EVA provide the optimal set from step*** $|\boldsymbol{u}|$ ***to the last step.*** *With* $\boldsymbol{u}$ *the essential variables of* $\boldsymbol{\delta}^*$*, EVA will rank the* $\boldsymbol{u}$ *variables first and provide the optimal set from the step* $|\boldsymbol{u}|$ *to the last step.*

Proof. Let $\boldsymbol{u}$ denote the *essential variables* of $\boldsymbol{\delta}^*$ and $\boldsymbol{v}$ the *inessential variables*. Then according to Proposition C.4 and Proposition C.5, $\forall i \in \boldsymbol{u}, \forall j \in \boldsymbol{v} : \text{EVA}(i) > \text{EVA}(j)$. It follow that $\boldsymbol{u}$ are the most important variables at step $|\boldsymbol{u}|$. Finally, according to Proposition C.3, $\boldsymbol{u}$ is the optimal set for k, with $k \in [\![|\boldsymbol{u}|, d]\!]$. □

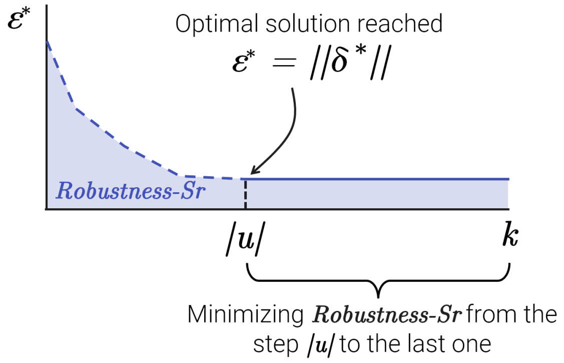
Optimal solution reached
$\mathcal{E}^* = //\delta^*//$
$\mathcal{E}^*$
Robustness-Sr
/u/
k
Minimizing Robustness-Sr from the step /u/ to the last one

Figure 6. **EVA yield optimal subset of variable from step $|\boldsymbol{u}|$.** $Robustness\text{-}S_{\text{r}}$ measures the AUC of the distances to the nearest adversary for the k most important variables. With $\boldsymbol{\delta}^*$ the nearest reachable adversarial perturbation around $\boldsymbol{x}$, then EVA yield the optimal set – the variables allowing to reach the nearest adversarial example for a given cardinality – at least from $||\boldsymbol{u}|| \leq d$ step to the last one, $\boldsymbol{u}$ being the so-called essential variables.

D. EVA and *Stability*

Stability is one of the most crucial properties of an explanation. Several metrics have been proposed [7, 69] and the most common one consists in finding around a point $\boldsymbol{x}$, another point $\boldsymbol{z}$ (in a radius r) such that the explanation changes the most according to a given distance between explanation d and a distance over the inputs ρ:

$$\textit{Stability}(\boldsymbol{x}, \boldsymbol{g}) = \max_{\boldsymbol{z}: \rho(\boldsymbol{z}, \boldsymbol{x}) \leq r} d(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{z}))$$

and $\boldsymbol{g}$ an explanation functional. It can be shown that the proposed EVA estimator is bounded by the stability of the model as well as by the radii ε and r, ε being the radius of $\mathcal{B}$ and r the radius of stability. From here, we assume d and ρ are the ℓ_2 distance.

Let assume that $\boldsymbol{f}$ is L-lipschitz. We recall that a function $\boldsymbol{f}$ is said L-lipschitz over $\mathcal{X}$ if and only if $\forall (\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{X}^2, ||\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{z})|| \leq ||\boldsymbol{x} - \boldsymbol{z}||$.

Theorem D.1. ***EVA has bounded Stability*** *Given a L-lipschitz predictor* $\boldsymbol{f}$*,* ε *the radius of* $\mathcal{B}$ *and* r *the Stability radius, then*

$$\textit{Stability}(\boldsymbol{x}, \textit{EVA}) \leq 4L(\varepsilon + r)$$

Proof. With $c' \neq c$ we denote $\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{f}_{c'}(\boldsymbol{x}) - \boldsymbol{f}_c(\boldsymbol{x})$. We note that by additivity of the Lipschitz constant $\boldsymbol{m}$ is $2L$-Lipschitz.

$$\begin{aligned}&\textit{Stability}(\boldsymbol{x}, \text{EVA}) = \max_{\boldsymbol{z}: \rho(\boldsymbol{z}, \boldsymbol{x}) \leq r} ||\text{EVA}(\boldsymbol{x}), \text{EVA}(\boldsymbol{z})|| \\ &= \max_{\boldsymbol{z}: \rho(\boldsymbol{z}, \boldsymbol{x}) \leq r} || \max_{\boldsymbol{\delta}} \boldsymbol{m}(\boldsymbol{x} + \boldsymbol{\delta}) - \max_{\boldsymbol{\delta}_{\boldsymbol{u}}} \boldsymbol{m}(\boldsymbol{x} + \boldsymbol{\delta}_{\boldsymbol{u}}) \\ &\quad - \max_{\boldsymbol{\delta}} \boldsymbol{m}(\boldsymbol{z} + \boldsymbol{\delta}) + \max_{\boldsymbol{\delta}_{\boldsymbol{u}}} \boldsymbol{m}(\boldsymbol{z} + \boldsymbol{\delta}_{\boldsymbol{u}})|| \\ &\leq \max_{\boldsymbol{z}: \rho(\boldsymbol{z}, \boldsymbol{x}) \leq r} || \max_{\boldsymbol{\delta}} \boldsymbol{m}(\boldsymbol{x} + \boldsymbol{\delta}) - \max_{\boldsymbol{\delta}} \boldsymbol{m}(\boldsymbol{z} + \boldsymbol{\delta})|| \\ &\quad + || \max_{\boldsymbol{\delta}_{\boldsymbol{u}}} \boldsymbol{m}(\boldsymbol{z} + \boldsymbol{\delta}_{\boldsymbol{u}}) - \max_{\boldsymbol{\delta}_{\boldsymbol{u}}} \boldsymbol{m}(\boldsymbol{x} + \boldsymbol{\delta}_{\boldsymbol{u}})|| \\ &= \max_{\boldsymbol{\gamma}: ||\boldsymbol{\gamma}|| \leq r} || \max_{\boldsymbol{\delta}} \boldsymbol{m}(\boldsymbol{x} + \boldsymbol{\delta}) - \max_{\boldsymbol{\delta}} \boldsymbol{m}(\boldsymbol{x} + \boldsymbol{\delta} + \boldsymbol{\gamma})|| \\ &\quad + || \max_{\boldsymbol{\delta}_{\boldsymbol{u}}} \boldsymbol{m}(\boldsymbol{x} + \boldsymbol{\delta}_{\boldsymbol{u}} + \boldsymbol{\gamma}) - \max_{\boldsymbol{\delta}_{\boldsymbol{u}}} \boldsymbol{m}(\boldsymbol{x} + \boldsymbol{\delta}_{\boldsymbol{u}})|| \\ &\leq 2L(||\boldsymbol{\delta}|| + ||\boldsymbol{\gamma}||) + 2L(||\boldsymbol{\delta}|| + ||\boldsymbol{\gamma}||) \\ &= 4L(\varepsilon + r)\end{aligned}$$

□

E. Attribution methods

In the following section, we give the formulation of the different attribution methods used in this work. The library used to generate the attribution maps is Xplique [18]. By simplification of notation, we define $\boldsymbol{f}(\boldsymbol{x})$ the logit score (before softmax) for the class of interest (we omit c). We recall that an attribution method provides an importance score for each input variable x_i. We will denote the explanation functional mapping an input of interest $\boldsymbol{x} = (x_1, ..., x_d) \in \mathcal{X}$ as $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^d$.

Saliency [56] is a visualization technique 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.

$$\boldsymbol{g}(\boldsymbol{x}) = ||\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})||$$

Gradient ⊙ Input [55] 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 ⊙ Input is equivalent to ϵ-LRP and DeepLIFT [55] methods under certain conditions – using a baseline of zero, and with all biases to zero.

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} \odot ||\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})||$$

Integrated Gradients [65] consists of summing the gradient values along the path from a baseline state to the current value. The baseline $\boldsymbol{x}_0$ used is 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 [62] for a comparison). For all the experiments $m = 100$.

$$\boldsymbol{g}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_0) \int_0^1 \nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}_0 + \alpha(\boldsymbol{x} - \boldsymbol{x}_0)))d\alpha$$

SmoothGrad [61] 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 an isotropic 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. The attribution is obtained by averaging after sampling m points. For all the experiments, we took $m = 100$ and $\sigma = 0.2 \times (\boldsymbol{x}_{\max} - \boldsymbol{x}_{\min})$ where $(\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max})$ being the input range of the dataset.

$$\boldsymbol{g}(\boldsymbol{x}) = \underset{\boldsymbol{\delta} \sim \mathcal{N}(0, \boldsymbol{I}\sigma)}{\mathbb{E}} (\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{\delta}))$$

VarGrad [28] is similar to SmoothGrad as it employs the same methodology to construct the attribution maps: using a set of m noisy inputs, it aggregates the gradients using the variance rather than the mean. For the experiment, m and σ are the same as Smoothgrad. Formally:

$$\boldsymbol{g}(\boldsymbol{x}) = \underset{\boldsymbol{\delta} \sim \mathcal{N}(0, \boldsymbol{I}\sigma)}{\mathbb{V}} (\nabla_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{\delta}))$$

Grad-CAM [53] can only be used on Convolutional Neural Network (CNN). Thus we couldn't use it for the MNIST dataset. The method 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 α_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, with $\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{A}_{ij}^k}$ and

$$\boldsymbol{g} = \max(0, \sum_k \alpha_k^c \boldsymbol{A}^k)$$

As the size of the explanation depends on the size (width, height) of the last feature map, a bilinear interpolation is performed in order to find the same dimensions as the input. For all the experiments, we used the last convolutional layer of each model to compute the explanation.

Grad-CAM++ (G+) [10] 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 :

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

Occlusion [71] is a sensitivity method that sweeps a patch that occludes pixels over the images using a baseline state and use the variations of the model prediction to deduce critical areas. For all the experiments, we took a patch size and a patch stride of $\frac{1}{7}$ of the image size. Moreover, the baseline state $\boldsymbol{x}_0$ was zero.

$$\boldsymbol{g}(\boldsymbol{x})_i = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}_{[\boldsymbol{x}_i = 0]})$$

RISE [48] is a black-box method that consists of probing the model with N randomly masked versions of the input image to deduce the importance of each pixel using the corresponding outputs. The masks $\boldsymbol{m} \sim \mathcal{M}$ are generated randomly in a subspace of the input space. For all the experiments, we use a subspace of size 7×7, $N = 6000$, and $\mathbb{E}(\mathcal{M}) = 0.5$.

$$\boldsymbol{g}(\boldsymbol{x}) = \frac{1}{\mathbb{E}(\mathcal{M})N} \sum_{i=0}^{N} \boldsymbol{f}(\boldsymbol{x} \odot \boldsymbol{m}_i) \boldsymbol{m}_i$$

Greedy-AS [29] is a greedy-like method which aggregates step by step the most important pixels – the pixels that allow us to obtain the closest possible adversarial example. Starting from an empty set, we evaluate the importance of the variables at each step. Formally, with $\boldsymbol{u}$ the feature set chosen at the current step and $\overline{\boldsymbol{u}}$ his complement. We define $b : \mathcal{P}(\overline{\boldsymbol{u}}) \rightarrow \{0, 1\}^{|\overline{\boldsymbol{u}}|}$ a function which binarizes a sub-set of the unchosen elements. Then, given the set of selected elements $\boldsymbol{u}$, we find the importance of the elements still not selected, while taking into account their interactions. This amounts to solving the following regression problem:

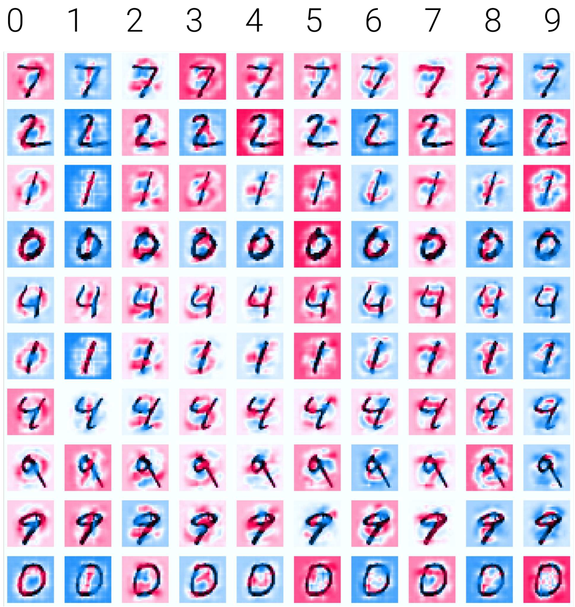

Figure 7. **Targeted Explanations** Attribution-generated explanations for a decision other than the one predicted. Each column represents the class explained, e.g., the first column looks for an explanation for the class '0' for each of the samples. As indicated in section 4.3, the red areas indicate that a black line should be added and the blue areas that it should be removed. More examples are available in the Appendix.

$$\boldsymbol{w}^t, c^t = \arg\min \sum_{\boldsymbol{v} \in \mathcal{P}(\overline{\boldsymbol{u}})} \left((\boldsymbol{w}^t b(\boldsymbol{v}) + c) - v(\boldsymbol{u} \cup \boldsymbol{v}) \right)^2$$

The weights obtained indicate the importance of each variable by taking into account these interactions. We specify that $v(\cdot)$ is defined here as the minimization of the distance to the nearest adversarial example using the variables $\boldsymbol{u} \cup \boldsymbol{v}$. In the experiments, the minimization of this objective is approximated using PGD [44] adversarial attacks, a regression step (computation of $\boldsymbol{w^t}$) adds 10% of the variables and $\boldsymbol{v}$ is sampled using 1000 samples from $\mathcal{P}(\boldsymbol{u})$. Finally, the variables added first to get a better score.

F. Evaluation

For the purpose of the experiments, three fidelity metrics have been chosen. For the whole set of metrics, $\boldsymbol{f}(\boldsymbol{x})$ score is the score after the softmax of the models.

Deletion. [48] The first metric is Deletion, it consists in measuring the drop in the score when the important variables are set to a baseline state. Intuitively, a sharper drop indicates that the explanation method has well-identified the important variables for the decision. The operation is repeated on the whole image until all the pixels are at a baseline state. Formally, at step k, with $\boldsymbol{u}$ the most important variables according to an attribution method, the Deletion$^{(k)}$ score is given by:

$$\text{Deletion}^{(k)} = \boldsymbol{f}(\boldsymbol{x}_{[\boldsymbol{x}_{\boldsymbol{u}} = \boldsymbol{x}_0]})$$

We then measure the AUC of the Deletion scores. For all the experiments, and as recommended by [29], the baseline state is not fixed but is a value drawn on a uniform distribution $\boldsymbol{x}_0 \sim \mathcal{U}(0, 1)$.

Insertion. [48] Insertion consists in performing the inverse of Deletion, starting with an image in a baseline state and then progressively adding the most important variables. Formally, at step k, with $\boldsymbol{u}$ the most important variables according to an attribution method, the Insertion$^{(k)}$ score is given by:

$$\text{Insertion}^{(k)} = \boldsymbol{f}(\boldsymbol{x}_{[\boldsymbol{x}_{\overline{\boldsymbol{u}}} = \boldsymbol{x}_0]})$$

The baseline is the same as for Deletion.

μ**Fidelity** [7] consists in measuring the correlation between the fall of the score when variables are put at a baseline state and the importance of these variables. Formally:

$$\mu\text{Fidelity} = \underset{\substack{\boldsymbol{u} \subseteq \{1, \ldots, d\} \\ |\boldsymbol{u}| = k}}{\text{Corr}} \left(\sum_{i \in \boldsymbol{u}} \boldsymbol{g}(\boldsymbol{x})_i, \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}_{[\boldsymbol{x}_{\boldsymbol{u}} = \boldsymbol{x}_0]}) \right)$$

For all experiments, k is equal to 20% of the total number of variables and the baseline is the same as the one used by Deletion.

G. Models

The models used were all trained using Tensorflow [1]. For MNIST, the model is a stacking of 5 dense layers composed of (256, 128, 64, 32, 10) neurons respectively. It achieves an accuracy score above 98% on the test set. Concerning the Cifar-10 model, it is composed of 3 Convolutional layers of (128, 80, 64) filters, a MaxPooling (2, 2), and to Dense layer of (64, 10) neurons respectively, and achieves 75% of accuracy on the test set. For ImageNet, we used a pre-trained VGG16 [57].

H. Targeted explanations

In order to generate targeted explanations, we split the calls to EVA$(\cdot, \cdot)$ in two: the first one with 'positive' perturbations from $\mathcal{B}^{(+)}$ (only positive noise), a call with 'negative' perturbations from $\mathcal{B}^{(-)}$ (only negative-valued noise) as defined in Section 4.3.

We then get two explanations, one for positive noise $\boldsymbol{\phi}^{(+)}_{\boldsymbol{u}} = \boldsymbol{F}_c(\mathcal{B}^{(+)}(\boldsymbol{x})) - \boldsymbol{F}_c(\mathcal{B}^{(+)}_{\boldsymbol{u}}(\boldsymbol{x}))$, the other for negative noise $\boldsymbol{\phi}^{(-)}_{\boldsymbol{u}} = \boldsymbol{F}_c(\mathcal{B}^{(-)}(\boldsymbol{x})) - \boldsymbol{F}_c(\mathcal{B}^{(-)}_{\boldsymbol{u}}(\boldsymbol{x}))$. Intuitively, high importance for $\boldsymbol{\phi}^{(+)}_{\boldsymbol{u}}$ means that the model is sensitive to the addition of a white line. Conversely, high importance for $\boldsymbol{\phi}_{\boldsymbol{u}}^{(-)}$ means that removing it changes the decision model. These two explanations being opposed, we construct the final explanation as $\boldsymbol{\phi}_{\boldsymbol{u}} = \boldsymbol{\phi}_{\boldsymbol{u}}^{(+)} - \boldsymbol{\phi}_{\boldsymbol{u}}^{(-)}$. More examples of results are given in Fig. 7.