Don't Lie to Me! Robust and Efficient Explainability with Verified Perturbation Analysis

Thomas Fel [*1] **Melanie Ducoffe** [*2] **David Vigouroux** [2] **Remi Cadene** [1] **Mikael Capelle** [2] **Claire Nicodeme** [3] **Thomas Serre** [1]

Abstract

A variety of methods have been proposed to try to explain how deep neural networks make their decisions. Key to those approaches is the need to sample the pixel space efficiently in order to derive importance maps. However, it has been shown that the sampling methods used to date introduce biases and other artifacts, leading to inaccurate estimates of the importance of individual pixels and severely limit the reliability of current explainability methods. Unfortunately, the alternative – to exhaustively sample the image space is computationally prohibitive. In this paper, we introduce EVA (Explaining using Verified perturbation Analysis) – the first explainability method guarantee to have an exhaustive exploration of a perturbation space. Specifically, we leverage the beneficial properties of verified perturbation analysis – time efficiency, tractability and guaranteed complete coverage of a manifold – to efficiently characterize the input variables that are most likely to drive the model decision. We evaluate the approach systematically and demonstrate state-of-the-art results on multiple benchmarks.

1. Introduction

Deep neural networks are nowadays widely used in many applications from medicine, transportation, security or finance, with broad societal implications (LeCun et al., 2015). Yet, these networks have become almost impenetrable. Plus, in most real-world applications, these systems are used to make critical decisions, often without any explanation.

[*]Equal contribution [1]Carney Institute for Brain Science, Brown University, USA [2]Artificial and Natural Intelligence Toulouse Institute, France [3]Innovation & Research Division, SNCF. Correspondence to: Firstname1 Lastname1 <first1.last1@xxx.edu>, Firstname2 Lastname2 <first2.last2@www.uk>.

1st Workshop on Formal Verification of Machine Learning, Baltimore, Maryland, USA. Colocated with ICML 2022.

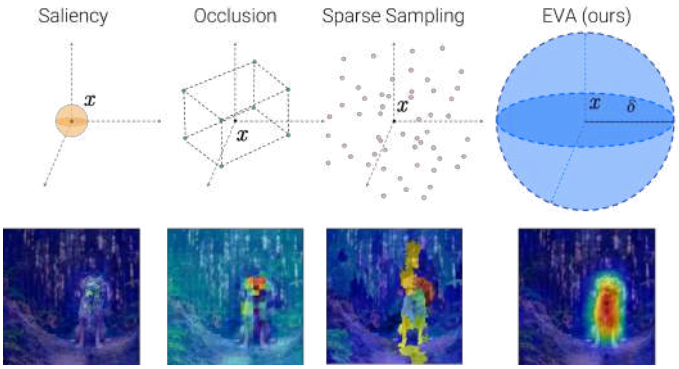

Figure 1. **Manifold exploration of current attribution methods.** Current attribution methods provide a notion of pixel importance using perturbations around an input $\boldsymbol{x}$. Saliency (Simonyan et al., 2013) uses infinitesimal perturbations around $\boldsymbol{x}$, Occlusion (Zeiler & Fergus, 2014a) simply varies each variables one by one towards a baseline state. Finally, several methods (Ribeiro et al., 2016; Lundberg & Lee, 2017; Petsiuk et al., 2018; Fel et al., 2021) use (Quasi-) random sampling in specific perturbation spaces. However, the choice of the perturbation space undoubtedly biases the results – potentially even introducing serious artifacts. We propose to use verified perturbation analysis to efficiently and systematically explore a perturbation space around $\boldsymbol{x}$ to generate reliable and faithful explanations.

Recently, a multitude of explainability methods have been advanced to provide insight into how models arrive at a particular decision (Zeiler & Fergus, 2014a; Ribeiro et al., 2016; Lundberg & Lee, 2017; Smilkov et al., 2017; Shrikumar et al., 2017; Sundararajan et al., 2017; Petsiuk et al., 2018; Selvaraju et al., 2017; Fel et al., 2021). These methods also aim to improve or debug the predictors – such as neural network – and, more importantly, to help instill some trust in the users of these systems (Doshi-Velez & Kim, 2017). Unfortunately, some methods exhibit severe limitations. In particular, they are subject to confirmation bias: while some methods appear to offer useful explanations to a human experimenter, they turn out not to reflect the actual behavior of the predictor (Adebayo et al., 2018; Ghorbani et al., 2017; Slack et al., 2021a). In addition, it has been shown that some commonly used benchmarks are biased, leading explainability methods to exploit these biases rather

than seeking to explain the predictions correctly (Sturmfels et al., 2020; Hsieh et al., 2021). For example, some fidelity metrics remove a variable – setting it to a baseline state – and look at the difference in prediction once the variable is removed to assess its importance It turns out that according to those metrics, a pixel being already at a baseline state does not change, neither does the prediction score. Thus, the metric will never assign importance score to a pixel that is already at a baseline state, while it is reasonable to think that it can be determinant for a prediction. More generally, for any baseline value, one obtains a metric biased towards this value, and this bias can be exploited by explainability methods. Hence, instead of providing confidence in the decisions of a system, these explanations may themselves be potentially flawed and unreliable.

A growing literature is therefore beginning to focus on the study of changes in model decisions based on adversarial perturbations (Hsieh et al., 2021; Boopathy et al., 2020; Lin et al., 2019; Ross et al., 2021). The explanation is then based on the robustness of the model. Specifically, a pixel or a region is considered important if it allows to easily generate an adversarial example – and thus makes the decision model change. In other words, it is an evaluation of the robustness of the system to specific image regions. This led to the design of several new robustness metrics to evaluate the quality of explanations. In order to optimize these new metrics, methods making intensive use of adversarial attacks have been proposed. However, current methods can be very computationally costly – sometimes more than 50000 tunable adversarial attacks per explanation – which makes their adoption in real cases complicated.

To meet these new objectives, we propose to use verified perturbation analysis, a rapidly growing field that develops methods to obtain bounds on the outputs of neural networks in the presence of input perturbations. Moreover, in contrast to current attributions methods based on on gradient or sampling, verified perturbation analysis allow to fully explore a perturbation space, see Figure 1.

Specifically, in this work, we introduce EVA (Explaining using Verified perturbation Analysis), a new explainability method based on robustness analysis. The method uses a tractable certified upper bound of robustness confidence thanks to verified perturbation analysis to derive a new estimator quantifying the importance of variables. Specifically, we identify the input variables that matter the most. That is, the variables most likely to change the predictor's decision (as opposed to simply changing the output units without necessarily affecting the class prediction). Using a thorough evaluation of several images datasets, we show that our method obtain convincing results on a large range of explainability metrics and that it is possible to use it on state-of-the-art models. Finally, we demonstrate that we can use the proposed method to generate class-specific explanations, and we study the effects of several verified perturbation analysis methods as an hyperparameter of the generated explanations.

2. Related Work

Attribution Methods. Our work builds on prior work aiming to develop attribution methods in order to explain the prediction of a deep neural network by pointing out to input variables that support the prediction (typically pixels or image regions for images – which lead to importance maps shown in Figure 1). The first method, Saliency, was introduced in (Baehrens et al., 2010). It was later refined in (Simonyan et al., 2014; Zeiler & Fergus, 2014b; Springenberg et al., 2014; Sundararajan et al., 2017; Smilkov et al., 2017) in the context of deep convolutional networks for classification. It consists in calculating a gradient derived from a classification score with respect to the input pixels using the backpropagation algorithm. However, the gradient only reflects the model's operation in an infinitesimal neighborhood around an input and can therefore be misleading (Ghalebikesabi et al., 2021). Other methods rely on perturbations and measure the difference in classification with the original image to produce an importance map Methods such as "Occlusion" (Zeiler & Fergus, 2014b), LIME (Ribeiro et al., 2016), RISE (Petsiuk et al., 2018), or Sobol (Fel et al., 2021) that leverage different sampling strategies to explore the space of perturbations around the image. For instance, Occlusion uses binary masks to occlude individual image regions, one at a time. RISE combines these discrete masks to perturb multiple regions at a time. Sobol uses continuous masks for a finer exploration of the space.

We argue that all these methods explore the space of perturbations in a sparse manner. They do not provide strong guarantees on the stability of the model's decision in the neighborhood of the points in the space that they explore. As a result, the explanations that they produce are not robust and might not be trustworthy. Instead, our method provides strong guarantees derived from the verified perturbation analysis. It allows for an efficient and exhaustive exploration of the space of perturbations.

Robustness based Explanation. In response to the many problems cited, several works (Ignatiev et al., 2019a;b; Slack et al., 2021b; Hsieh et al., 2021; Boopathy et al., 2020; Lin et al., 2019; Fel & Vigouroux, 2022) have proposed a new set of robustness-based evaluation criteria for trustworthy explanations. These criteria are opposed to evaluations that are based on the removal of features that inevitably introduce biases and artifacts (Hsieh et al., 2021). These new guidelines are mainly based on the following assumption: when the important variables are in their nominal (fixed) state, then perturbations on the complementary variables –

deemed unimportant – should not affect the model's decision to any great extent. The corollary that follows is that perturbations limited to the variables considered as important should easily influence the model's decision (Lin et al., 2019; Hsieh et al., 2021). Based on these assumptions, the authors of (Hsieh et al., 2021) proposed the $Robustness\text{-}S_r$ metric that quantifies the ability of an explanation to find the important variables. Moreover they argue that their method, unlike current metrics such as Deletion, Insertion (Petsiuk et al., 2018), is not affected by biases. They also propose to take a baseline to alleviate the impact of biases on the current metrics to improve them.

Some works then propose to optimize these criteria and propose new methods using a generative model (O'Shaughnessy et al., 2020) or adversarial attacks (Hsieh et al., 2021). This last approach requires checking the existence or not of an adversarial example for a multitude of ℓ_p balls around the input of interest. Therefore the induced computational cost is particularly high, as evidenced by the experiments which require more than 50000 computations of adversarial examples to generate a single explanation. More importantly, not finding an adversarial perturbation for a given radius does not guarantee that none exists. Moreover, it is not uncommon for adversarial attacks to fail to converge – to fail to find any adversarial example – thus not yielding any importance score. Our method addresses these issues while keeping the same objectives, by taking advantage of the certificates generated using verified perturbation analysis.

Verified Perturbation Analysis. Orthogonally, the growing field of Verified Perturbation Analysis aims to find methods that outer-approximate neural network outputs under input perturbations. Simply put, for a given input $\boldsymbol{x}$ and a bounded perturbation $\boldsymbol{\delta}$, the verification methods allow us to obtain a minimum $\underline{\boldsymbol{f}}(\boldsymbol{x})$ and a maximum $\overline{\boldsymbol{f}}(\boldsymbol{x})$ bound on the output of a model, Formally $\forall\ \boldsymbol{\delta}\ s.t\ ||\boldsymbol{\delta}||_p \leq \varepsilon$:

$$\underline{\boldsymbol{f}} \leq \boldsymbol{f}(\boldsymbol{x}+\boldsymbol{\delta}) \leq \overline{\boldsymbol{f}}$$

Thus allowing us to explore the whole perturbation set without having to explicitly sample all the points.

Early works in the domain focused on computing reachable lower and upper bounds based on satisfiability modulo theory (Katz et al., 2017; Ehlers, 2017), and mixed-integer linear programming problems (Tjeng & Tedrake, 2019). Despite this, they struggled to adapt to a small network, even on the smallest image dataset. Recently, many other researches have independently discovered how to compute looser certified lower and upper bounds more efficiently thanks to convex linear relaxations either in the primal or dual space (Salman et al., 2019). If looser, those bounds remain tight enough to prove non ubiquitous robustness properties on medium size neural networks. Among those methods, CROWN hereafter called Backward (Zhang et al., 2018; Singh et al., 2019; Wang et al., 2021) stands for the state-of-the-art method for Linear Relaxation based Perturbation Analysis (LiRPA), achieving the tightest bound for efficient single neuron linear relaxation. While CROWN is the most efficient, its polynomial complexity compared to the computational cost of inference limits its application to networks of the size of AlexNet (Krizhevsky et al., 2012). However, linear relaxation offers a wide range of possibilities with a vast trade-off between scalability and efficiency. These methods are declined in two forms. Firstly, those which propagate constant bounds, more generally affine relaxations from the input to the output of the network, also called Interval Bound Propagation (IBP, Forward, IBP+Forward). Conversely, the so-called 'backward' methods will bound the output of the network by affine relaxations given the internal layers of the network, starting from the output to the input. Note that these methods can be combined, e.g. (CROWN + IBP + Forward). For a thorough description of the underlying specificities of LiRPA's framework and a theoretical analysis of the worst case complexities of each methods, see (Xu et al., 2020). In this work, we take the approach of being agnostic to the verification method and opt for the most accurate LiRPA method applicable on the predictor. Our approach is based on the formal verification framework DecoMon, based on Keras (Ducoffe, Melanie, 2021).

3. Explainability with Verified Pertubation Analysis

3.1. Notation

We consider a standard supervised machine learning classification setting with input space $\mathcal{X} \subseteq \mathbb{R}^d$ an output space $\mathcal{Y} \subseteq \mathbb{R}^c$ and a predictor function $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$ that map an input vector $\boldsymbol{x} = (x_1, \ldots, x_d)$ to an output $\boldsymbol{f}(\boldsymbol{x}) = (\boldsymbol{f}_1(\boldsymbol{x}), \ldots, \boldsymbol{f}_c(\boldsymbol{x}))$. We denote $\mathcal{B} = \{\boldsymbol{\delta} \in \mathbb{R}^d : ||\boldsymbol{\delta}||_p \leq \varepsilon\}$ the perturbation ball with radius $\varepsilon > 0$, with $p \in \{1, 2, \infty\}$. For any subset of indices $\boldsymbol{u} \subseteq \{1, \ldots, d\}$, we denote $\mathcal{B}_{\boldsymbol{u}}$ the ball without perturbation on the variables in $\boldsymbol{u}$: $\mathcal{B}_{\boldsymbol{u}} = \{\boldsymbol{\delta} : \boldsymbol{\delta} \in \mathcal{B},\ \boldsymbol{\delta}_{\boldsymbol{u}} = 0\}$ and $\mathcal{B}(\boldsymbol{x})$ the perturbation ball centered on $\boldsymbol{x}$. We denote the lower (resp. upper) bounds obtained with verification perturbation analysis as $\underline{\boldsymbol{f}}(\boldsymbol{x}, \mathcal{B}) = \left(\underline{\boldsymbol{f}}_1(\boldsymbol{x}, \mathcal{B}), \ldots, \underline{\boldsymbol{f}}_c(\boldsymbol{x}, \mathcal{B})\right)$, and $\overline{\boldsymbol{f}}(\boldsymbol{x}, \mathcal{B}) = \left(\overline{\boldsymbol{f}}_1(\boldsymbol{x}, \mathcal{B}), \ldots, \overline{\boldsymbol{f}}_c(\boldsymbol{x}, \mathcal{B})\right)$. Intuitively, these bounds delimit the output prediction for any perturbed sample in $\mathcal{B}(\boldsymbol{x})$.

3.2. The importance of setting the importance

The goal of an attribution method being to assign an importance score to each variable, we can deduce a definition of importance for each existing method. For example, this

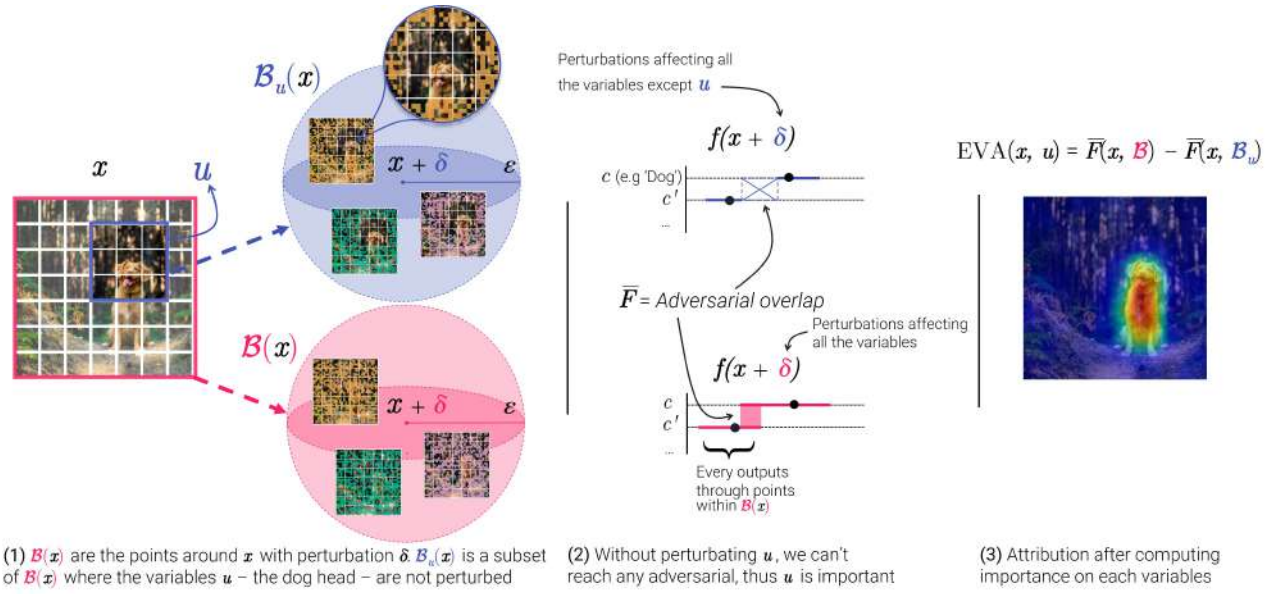

Figure 2. **EVA attribution method.** In order to compute the importance for a group of variables $\boldsymbol{u}$ – for instance the dog's head – the first step (1) consists in forming the perturbation ball $\mathcal{B}_u(\boldsymbol{x})$. This ball is centered in $\boldsymbol{x}$ and contain all the possible images perturbed by δ $s.t$ $||\boldsymbol{\delta}||_p \leq \varepsilon, ||\boldsymbol{\delta}_{\boldsymbol{u}}||_p = 0$ which do not perturb the variables $\boldsymbol{u}$. Using verified perturbation analysis, we then compute the *adversarial overlap* $\overline{\boldsymbol{F}}(\boldsymbol{x}, \mathcal{B}_u)$ which corresponds to the overlapping between the class c – here dog – and c', the maximum among the other classes. Finally, the importance score for the variable $\boldsymbol{u}$ corresponds to the drop in *adversarial overlap* when $\boldsymbol{u}$ cannot be perturbed, thus the difference between $\boldsymbol{F}(\boldsymbol{x}, \mathcal{B})$ and $\boldsymbol{F}(\boldsymbol{x}, \mathcal{B}_u)$. Specifically, this measures how important the variables $\boldsymbol{u}$ are for changing the model's decision.

definition can be based on game theory (Lundberg & Lee, 2017), on the conditional expectation of score logits (Petsiuk et al., 2018), or on its variance (Fel et al., 2021). In this work, we consider – like (Hsieh et al., 2021) – that a variable is important if, when modified, it can change the model decision. Conversely, a variable is said unimportant if its modification does not modify the decision. From this, we go one step further than the previous work and propose to couple the ability to change decision with other information, notably confidence in the prediction, to evaluate importance. Based on those considerations, we derive a score to quantify classes' overlap – how much the maximum attainable of an adverse class can concur with the minimum of the initial class – that we call the *adversarial overlap*. We then use this criterion to build our importance estimator.

3.3. *Adversarial overlap*

In order to determine the importance score from the previous motivation, we rely on the capacity of a variable to change the decision of the model. Indeed, if the manipulation of a variable allows to generate a new input that alters the decision, this variable is of interest. Conversely, if the decision does not change whatever its state, the variable can be left at its nominal value. Among the set of possible variable perturbations $\boldsymbol{\delta}$ around a point $\boldsymbol{x}$, we therefore look for points that can modify the decision with most confidence. Hence our scoring criterion can be formulated as follows

$$\boldsymbol{F}_c(\boldsymbol{x}, \mathcal{B}) = \max_{\substack{\boldsymbol{\delta} \in \mathcal{B} \\ c' \neq c}} \boldsymbol{f}_{c'}(\boldsymbol{x} + \boldsymbol{\delta}) - \boldsymbol{f}_c(\boldsymbol{x} + \boldsymbol{\delta}). \tag{1}$$

Intuitively, this score represents the confidence of the "best" adversarial perturbation that can be found in the perturbation ball $\mathcal{B}$ around $\boldsymbol{x}$. In order to estimate this criterion, a first method could be to use adversarial attacks to search within $\mathcal{B}$. However, such methods only explore certain points of the considered space, thus giving no guarantee on the optimality of the solution. Moreover, adversarial methods have no guarantee of success and therefore cannot ensure a valid score under all circumstances. Finally, the large dimensions of the current datasets prevents the possibility of exhaustive searches.

To overcome these issues, we take advantage of verified perturbation analysis to obtain a guaranteed upper bound on the criterion introduced in Equation 1. We can upper bound the *adversarial overlap* criterion as the following:

$$\boldsymbol{F}_c(\boldsymbol{x}, \mathcal{B}) \leq \bar{\boldsymbol{F}}_c(\boldsymbol{x}, \mathcal{B}) = \max_{c' \neq c} \overline{\boldsymbol{f}}_{c'}(\boldsymbol{x}, \mathcal{B}) - \underline{\boldsymbol{f}}_c(\boldsymbol{x}, \mathcal{B})$$

which then becomes tractable by any verified perturbation analysis method.

For example, in Figure 3, $\bar{\boldsymbol{F}}(\boldsymbol{x}, \mathcal{B}) \leq 0$ guarantees that no adversarial perturbation is possible in the perturbation

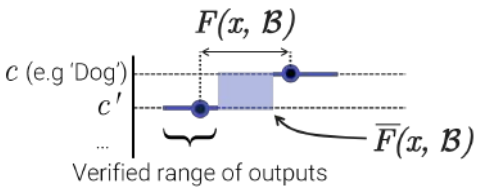

Figure 3. Illustration of verified perturbation analysis for bounding *adversarial overlap*. The blue points correspond respectively from left to right to $\boldsymbol{f}_{c'}(\boldsymbol{x}+\boldsymbol{\delta})$, $\boldsymbol{f}_{c}(\boldsymbol{x}+\boldsymbol{\delta})$ such that $\boldsymbol{F}_c(\boldsymbol{x}, \mathcal{B}) = \boldsymbol{f}_{c'}(\boldsymbol{x}+\boldsymbol{\delta}) - \boldsymbol{f}_c(\boldsymbol{x}+\boldsymbol{\delta})$. Verified perturbation analysis bounds the output prediction for any perturbed points with upper and lower bounds. From these bounds, we can deduce an over-approximation of the *adversarial overlap* score $\bar{\boldsymbol{F}}_c(\boldsymbol{x}, \mathcal{B})$.

space. Note that with adversarial attacks, failure to find an adversarial example does not guarantee that it does not exist. Our upper bound $\bar{\boldsymbol{F}}_c(\boldsymbol{x}, \mathcal{B})$ corresponds to the difference between the verified lower bound of the class of interest c and the maximum over the verified upper bounds among the other classes. Thus, when important variables are modified (e.g the head of the dog in Figure 2, using $\mathcal{B}$), the lower bound of the class of interest will get smaller than the upper bound of an adversary class. On the other hand, this overlap is not possible when important variables are fixed (e.g in Figure 2 when the head of the dog is fixed, using $\mathcal{B}_{\boldsymbol{u}}$). We now demonstrate how to leverage this score to derive an efficient estimator of variable importance.

3.4. EVA – Explaining using Verified perturbation Analysis

We are willing to assign a higher importance score for a variable allowing (1) a change in a decision, (2) a greater adversarial – thus a solid change of decision. Modifying all variables gives us an idea of the robustness of the model. In the same way, the modification of all variables without the subset $\boldsymbol{u}$ allows to quantify the change of the strongest adversarial perturbation and thus to quantify the importance of the variables $\boldsymbol{u}$. Intuitively, if an important variable $\boldsymbol{u}$ is discarded, then it will be more difficult, if not impossible, to succeed in finding any adversarial perturbation. Specifically, removing the possibility to modify $\boldsymbol{x}_{\boldsymbol{u}}$ allows us to reveal its importance by taking into account its possible interactions.

The complexity of current models means that the variables are not only treated individually in neural network models but collectively. In order to capture these higher order interactions, our method consists in measuring the *adversarial overlap* allowed by all the variables together $\boldsymbol{F}(\boldsymbol{x}, \mathcal{B})$ – thus taking into account their interactions – and then forbidding to play on a group of variables $\boldsymbol{F}(\boldsymbol{x}, \mathcal{B}_{\boldsymbol{u}})$ to estimate the importance of $\boldsymbol{u}$. Making the interactions of $\boldsymbol{u}$ disappear reveals their importance. Note that several works have mentioned the importance of taking into account the interactions of the variables when calculating the importance (Petsiuk et al., 2018; Fel et al., 2021; Ferrettini et al., 2021). Formally, we introduce EVA (Explainability using Verified perturbation Analysis) that measure the drop in *adversarial overlap* when we fixed the variables $\boldsymbol{u}$:

$$\text{EVA}(\boldsymbol{x}, \boldsymbol{u}) = \bar{\boldsymbol{F}}(\boldsymbol{x}, \mathcal{B}) - \bar{\boldsymbol{F}}(\boldsymbol{x}, \mathcal{B}_{\boldsymbol{u}}) \tag{2}$$

As explained in Figure 2, the estimator requires two passes of the perturbation analysis method; one for $\bar{\boldsymbol{F}}(\mathcal{B})$, and the other for $\bar{\boldsymbol{F}}(\mathcal{B}_{\boldsymbol{u}})$: the first term consists in measuring the *adversarial overlap* by modifying all the variables, the second term measures the adversarial surface when fixing the variables of interest $\boldsymbol{u}$. In other words, EVA measures the *adversarial overlap* that would be left if the variables $\boldsymbol{u}$ were to be fixed.

From a theoretical point of view, we notice that EVA - under some conditions - yield the optimal subset of variables to minimize the theoretical $Robustness\text{-}S_r$ metric (see Theorem A.6). From a computational point of view, we can note that the first term of the *adversarial overlap* $\boldsymbol{F}(\boldsymbol{x}, \mathcal{B})$ – as it does not depend on $\boldsymbol{u}$ – can be calculated once and re-used to evaluate the importance of any other variables considered. Moreover, contrary to an iterative process method (Fong & Vedaldi, 2017; Hsieh et al., 2021; Ignatiev et al., 2019a), each importance can be evaluated independently and thus benefit from the parallelization of modern neural networks. Finally, the experiments in Section 4 show that even with two calls to *adversarial overlap* per variables, our method remains much faster than the one based on adversarial attacks, see the results concerning the computing time in Table 1.

In this work, the verified perturbation based analysis considered are not always adapted to high dimensional models, especially those running on ImageNet (Deng et al., 2009). We are confident that the verification methods will progress towards more scalability in the near future, enabling the original version of EVA on deeper models. In the meantime, we introduce in the next section an empirical method to compute EVA on large models. We then show the interest of this method on the ImageNet dataset in the experiments.

3.5. Scaling strategy

In this section, we propose an empirical method to scale our method on ImageNet. Indeed, since verified perturbation analysis is not directly applicable on ImageNet models so far, we handle this scalability issue by combining empirical bounds on some hidden layers, and then composing these bounds with verified perturbation analysis on the last layers of the network. Our modification of EVA takes inspiration from the work of (Balunovic & Vechev, 2019) who introduced an hybrid robust learning scheme by combining empirical methods (a.k.a adversarial training) with a verified

	MNIST					Cifar-10					ImageNet			
	Del.↓	Ins.↑	Fid.↑	Rob.↓	Time	Del.↓	Ins.↑	Fid.↑	Rob.↓	Time	Del.↓	Ins.↑	Fid.↑	Rob.↓
Saliency	.193	.633	.378	.071	0.04	.171	.172	-.021	.026	0.16	.057	.126	.035	.769
GradInput	.222	.611	.107	.074	0.04	.200	.143	-.018	.095	0.17	.057	.050	.023	.814
SmoothGrad	.185	.621	.331	.070	1.91	.174	.181	.092	.048	9.07	**.051**	.069	.019	.809
VarGrad	.207	.555	.216	.077	1.76	.183	.211	-.012	.193	9.07	.098	.201	.021	.787
InteGrad	.209	.615	.108	.074	1.77	.194	.171	-.016	.154	7.19	.058	.052	.023	.813
Occlusion	.247	.545	.137	.082	0.04	.217	.290	.105	.232	1.13	.100	.266	.026	.821
GradCAM	na	na	na	na	na	.297	.282	.056	.195	0.39	.073	.232	.036	.817
GradCAM++	na	na	na	na	na	.270	**.326**	.102	.094	0.39	.074	.285	.054	.800
RISE	.248	.558	.133	.093	2.26	.196	.273	.157	.385	20.5	.074	.276	**.154**	.818
GreedyAS	.260	.497	.110	**.061**	335.8	.205	.264	-.003	**.013**	4618	.088	.047	.023	**.612**
EVA (ours)	**.089**	**.736**	**.428**	.069	1.29	**.164**	.290	**.352**	.025	12.7	.070	**.289**	.048	.758

Table 1. Results on Deletion (Del.), Insertion (Ins.), μFidelity (Fid.) and $Robustness\text{-}S_r$ (Rob.) metrics. The Time in seconds corresponds to the generation of 100 explanations on a Nvidia P100. For MNIST, the verified perturbation analysis used is (IBP + Forward + Backward), Forward is used for Cifar-10 and our empirical strategy is used for ImageNet. Grad-CAM and Grad-CAM++ are not calculated on the mnist dataset since the network has only dense layers. The first and second best results are respectively in **bold** and underlined.

perturbation-based analysis.

Specifically, our technique consists of splitting the model into two parts, and (1) estimating the bounds of an intermediate layer using sampling, (2) propagating these empirical intermediate bounds onto the second part of the model with verified perturbation analysis methods.

For the first step we consider a $\boldsymbol{f}$ as a l layers neural network $\boldsymbol{f} = \boldsymbol{h}^l \circ \boldsymbol{h}^{l-1} \circ \ldots \circ \boldsymbol{h}^1(\boldsymbol{x})$, we propose to empirically estimate bounds $(\underline{\boldsymbol{h}}^{i,\boldsymbol{x}}, \overline{\boldsymbol{h}}^{i,\boldsymbol{x}})$ for the intermediate state $\boldsymbol{h}^i(.) \in \mathbb{R}^{d'}$ with $1 \leq i < l$ using Monte-Carlo sampling on the perturbation $\boldsymbol{\delta} \in \mathcal{B}$. Obviously, since the sampling is never exhaustive, the bounds obtained underestimate the true maximum $\overline{\boldsymbol{h}}^{i,\boldsymbol{x}} \leq \max \boldsymbol{h}^i(\boldsymbol{x} + \boldsymbol{\delta})$ and overestimates the true minimum $\underline{\boldsymbol{h}}^{i,\boldsymbol{x}} \geq \min \boldsymbol{h}^i(\boldsymbol{x} + \boldsymbol{\delta})$. In a similar way, we define $\overline{\boldsymbol{h}}^{i,\boldsymbol{x},\boldsymbol{u}}$ and $\underline{\boldsymbol{h}}^{i,\boldsymbol{x},\boldsymbol{u}}$ for $\boldsymbol{\delta} \in \mathcal{B}_{\boldsymbol{u}}$.

Once the empirical bounds are estimated, we may proceed to the second step and use the obtained bounds to form the new perturbation set $\mathcal{P}^{i,\boldsymbol{x}}$ of all possible activations states on the i layer such that:

$$\mathcal{P}^{i,\boldsymbol{x}} = \{\boldsymbol{\delta} \in \mathbb{R}^{d'} \ : \ \underline{\boldsymbol{h}}_j^{i,\boldsymbol{x}} \leq \boldsymbol{h}^i(\boldsymbol{x})_j + \boldsymbol{\delta}_j \leq \overline{\boldsymbol{h}}_j^{i,\boldsymbol{x}}\}$$

and $\mathcal{P}_{\boldsymbol{u}}^{i,\boldsymbol{x}}$ the set of all possible perturbations of the activation states when the variable $\boldsymbol{u}$ was not affected by the perturbations:

$$\mathcal{P}_{\boldsymbol{u}}^{i,\boldsymbol{x}} = \{\boldsymbol{\delta} \in \mathbb{R}^{d'} \ : \ \underline{\boldsymbol{h}}_j^{i,\boldsymbol{x},\boldsymbol{u}} \leq \boldsymbol{h}^i(\boldsymbol{x})_j + \boldsymbol{\delta}_j \leq \overline{\boldsymbol{h}}_j^{i,\boldsymbol{x},\boldsymbol{u}}\}$$

We then carry out the end of the bounds propagation in the usual way, using verified perturbation analysis. This amounts to computing bounds for the outputs of the network for all possible activations contained in our empirical bounds. The only change being that we no longer operate in the pixel space $\boldsymbol{x}$ with the ball $\mathcal{B}$, but in the activation space $\boldsymbol{h}^i(\boldsymbol{x})$ with the perturbations set $\mathcal{P}^{i,\boldsymbol{x}}$. The importance score of a set of variables $\boldsymbol{u}$ is then :

$$\text{EVA}(\boldsymbol{x}, \boldsymbol{u}) = \overline{\boldsymbol{F}}(\boldsymbol{h}^i(\boldsymbol{x}), \mathcal{P}^{i,\boldsymbol{x}}) - \overline{\boldsymbol{F}}(\boldsymbol{h}^i(\boldsymbol{x}), \mathcal{P}_{\boldsymbol{u}}^{i,\boldsymbol{x}})$$

This empirical method allows to use EVA on state-of-the-art models and thus to benefit from our method while remaining tractable. We believe this extension to be a promising step towards robust explanations on deeper networks.

4. Experiments

To evaluate the benefits and reliability of our explainability method, we performed several experiments on standard dataset, using a set of common explainability metrics against EVA. In order to test the fidelity of the explanations produced by our method, we compared them to that of 10 other explainability methods using the (1) Deletion, (2) Insertion and (3) MuFidelity metrics. As it has been shown that these metrics can exhibit biases, we completed the benchmark by adding the (4) $Robustness\text{-}S_r$ metric. Each score is averaged over 500 samples.

We evaluated these 4 metrics on 3 image classification datasets. First, we conducted our experiments on MNIST (LeCun & Cortes, 2010) composed of 28x28 grayscale handwritten digit image. Then we experimented on CIFAR10 (Krizhevsky et al., 2009), a low-resolution labeled dataset with 10 classes composed of color image of resolution 32×32. Finally, we assessed the methods on ILSVRC 2012 (Deng et al., 2009), the test set of ImageNet dataset containing images of size 224×224.

Through these experiments, the explanations were generated using EVA estimator introduced in Equation 2. The

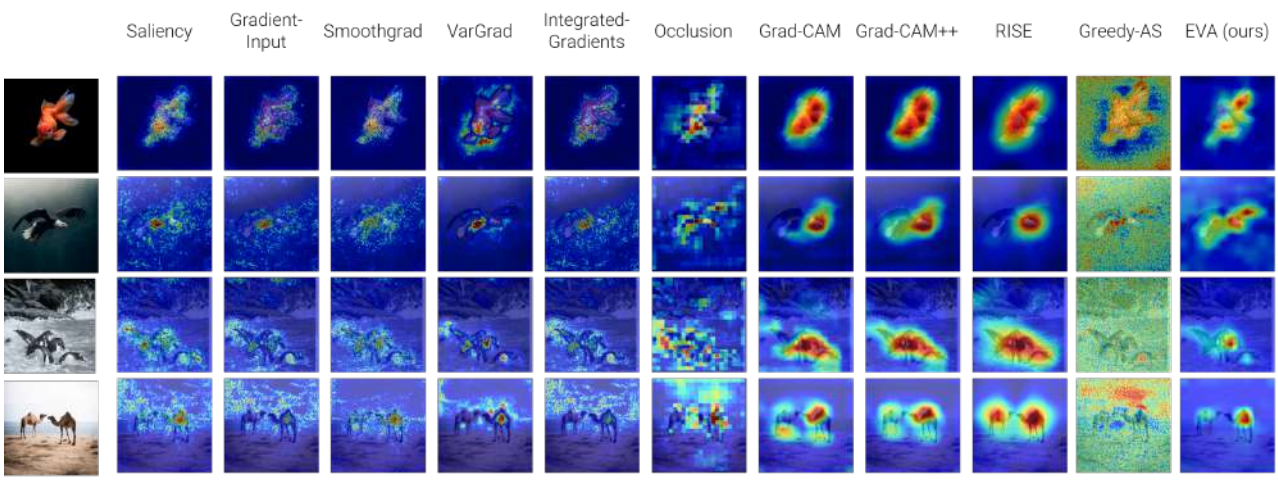

Figure 4. **Qualitative comparison** with other attribution methods. To allow better visualization, the gradient-based methods (Saliency, Gradient-Input, SmoothGrad, Integrated-Gradient, VarGrad) were 2 percentile clipped. For more results and details on each methods and hyperparameters, see the appendix.

importance scores were not evaluated pixel-wise but on each cell of the image after having cut it into a grid of 12 sides (see Figure 2). For MNIST and Cifar-10, we used $\varepsilon = 0.5$, whereas for ImageNet $\varepsilon = 5$. Concerning the verified perturbation analysis method, we used (IBP+Forward+Backward) for MNIST, and (IBP+Forward) on Cifar-10. For computational purposes, we tested the empirical strategy introduce in Section 3.5 for ImageNet using the penultimate layer (FC-4096) as the intermediate layer $\boldsymbol{h}^i(\cdot)$. We give in appendix the complete set of hyperparameters used for the other explainability methods, metrics considered as well as the architecture of the models used on MNIST and Cifar-10.

4.1. Fidelity & Robustness

There is a general consensus that fidelity is a crucial criterion for an explanation method. That is, if an explanation is used to make a critical decision, then users are expecting it to reflect the true decision-making process underlying the model and not just a consensus with humans. Failure to do so could have disastrous consequences. Pragmatically, these metrics assume that the more faithful an explanation is, the faster the prediction score should drop when pixels considered important are changed. In Table 1, we present the results of the Deletion (Petsiuk et al., 2018) (or $1 - AOPC$ (Samek et al., 2016)) metric for the MNIST and Cifar-10 datasets on 500 images sampled from the test set. TensorFlow (Abadi et al., 2015) and the Keras API (Chollet et al., 2015) were used to run the models. In order to evaluate the methods, the metrics require a baseline and several were proposed (Sturmfels et al., 2020; Hsieh et al., 2021), but we chose to keep the choice of (Hsieh et al., 2021) using their random baseline.

We observe that EVA is the explainability method getting the best Deletion, Insertion and μFidelity scores on MNIST, and is just behind Greedy-AS on $Robustness\text{-}S_r$. This can be explained by the fact that the Robustness metric uses the adversarial attack PGD (Madry et al., 2018), which is the same one used to generate Greedy-AS, thus biasing the adversarial search. Indeed, if PGD does not find an adversarial perturbation using a subset $\boldsymbol{u}$ does not give a guarantee on the robustness of the model, just that the adversarial perturbation could be difficult to reach with PGD.

For Cifar-10, EVA remains overall the most faithful method according to Deletion and μFidelity, and obtains the second score in Insertion behind Grad-Cam++ (Chattopadhay et al., 2018). Finally, we notice that if Greedy-AS (Hsieh et al., 2021) allows us to obtain a good $Robustness\text{-}S_r$ score, but this comes with a considerable computation time, which is not the case of EVA which is much more efficient. Eventually, EVA is a very good compromise for its relevance to commonly accepted explainability metrics and more recent robustness metrics.

ImageNet After having demonstrated the potential of the method on vision datasets of limited size, it is interesting to consider the case of ImageNet which has a significantly higher level of dimension. The use of verified perturbation analysis methods other than IBP is currently not able to scale on these datasets. We therefore used the empirical method introduced in Section 3.5 in order to estimate in a latent space the bounds and then plug those bounds into the perturbation analysis to get the final *adversarial overlap* score.

The Table 1 shows the results obtained with the empirical method proposed in Section 3.5. We observe that even with this relaxed estimation, EVA is able to score high on all the metrics. Indeed, EVA obtains the best score on the Insertion

Figure 5. **Targeted explanations** Attributions generated explanation for a decision other than the one predicted. The class explained is indicated at the bottom of each sample – e.g, the first sample is a '4' and the explanation is for the class '9'. As indicated in section 4.2, 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.

metric and ranks second on μFidelity and $Robustness\text{-}S_r$. Greedy-AS ranks first on $Robustness\text{-}S_r$ at the expense of the other scores where it performs poorly. Finally, Rise and SmoothGrad perform well on all the fidelity metrics but collapse on the robustness metric.

Qualitatively, Figure 4 shows examples of explanations produced on the ImageNet VGG-16 model. The explanations produced by EVA are more localized than Grad-CAM or RISE, while being less noisy than the gradient-based or Greedy-AS methods.

In addition, as the literature on verified perturbation analysis is evolving rapidly we can conjecture that the advances will benefit the proposed explainability method. Indeed, EVA proved to be the most effective on the benchmark when an accurate formal methods was used. After demonstrating the performance of the proposed method, we want to study its ability to generate explanations specific to each class.

4.2. Targeted Explanations

In some cases, it is instructive to look at the explanations for unpredicted classes in order to get information about the internal mechanisms of the models studied. Such explanations allow to highlight contrastive features: elements that should be changed or whose absence is critical. Our method allows us to obtain such explanations: for a given input, we are then exclusively interested in the class we are trying to explain, without looking at the other decisions. Formally, for a given targeted class c' the *adversarial overlap* (Equation 1) become $\boldsymbol{F}_c(\boldsymbol{x}, \mathcal{B}) = \max_{\boldsymbol{\delta}\in\mathcal{B}} \boldsymbol{f}_{c'}(\boldsymbol{x}+\boldsymbol{\delta}) - \boldsymbol{f}_c(\boldsymbol{x}+\boldsymbol{\delta})$. Moreover, by splitting the perturbation ball into a positive one $\mathcal{B}^{(+)} = \{\boldsymbol{\delta} \in \mathcal{B} \ : \ \boldsymbol{\delta}_i \geq 0, \ \forall i \in \{1, ..., d\}\}$ and a negative one $\mathcal{B}^{(-)} = \{\boldsymbol{\delta} \in \mathcal{B} \ : \ \boldsymbol{\delta}_i \leq 0, \ \forall i \in \{1, ..., d\}\}$, one can deduce which direction – adding or removing black line in the case of gray-scaled images – will impact the most the model decision.

We generated targeted explanations on the MNIST dataset using (IBP+Forward+Backward). For several inputs, we generate the explanation for the 10 classes. Figure 7 shows 4 examples of targeted explanations, the target class c' is indicated at the bottom. The red areas indicate that adding a black line increases the *adversarial overlap* with the target class. Conversely, the blue areas indicate where the increase of the score requires to remove black lines. All other results can be found in the appendix. In addition to favorable results on the fidelity metrics and guarantees provided by the verification methods, EVA can provide targeted explanations that are easily understandable by humans, which are two qualities that make them a candidate of choice to meet the recent General Data Protection Regulation (GDPR) adopted in Europe (Kaminski, 2021). More examples are available in the Appendix E.

4.3. Tighter bounds lead to improved explanations

	Tightness↓	Del.↓	Ins.↑	Fid.↑	Rob.↓
IBP	4.58	.148	.588	.222	.077
Forward	2.66	.150	.580	.209	.078
Backward	<u>2.36</u>	<u>.115</u>	<u>.607</u>	<u>.274</u>	<u>.074</u>
IBP + Fo. + Ba.	**1.55**	**.089**	**.736**	**.428**	**.069**

Table 2. **Impact of the verified perturbation analysis method on EVA.** Results of EVAon Tightness, Deletion (Del.), Insertion (Ins.), Fidelity (Fid.) and $Robustness\text{-}S_r$ (Rob.) metrics obtained on MNIST. The Tightness score corresponds to the average adversarial surface. A lower Tightness score indicates that the method is more precise: it reaches tighter bound, resulting in better explanations and superior scores on the other metrics. The first and second best results are respectively in **bold** and <u>underlined</u>.

Verified perturbation analysis method is an hyperparameter of EVA. Hence, it is interesting to see the effect of the choice of this method on the previous benchmark. We recall that only the MNIST dataset could benefit from the (IBP+Forward+Backward) combo. Table 2 reports the results of the fidelity metrics using other verified perturbation analysis methods. We also report a tightness score which corresponds to the average of the *adversarial overlap* : $\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{X}}(\boldsymbol{F}(\boldsymbol{x}, \mathcal{B}))$. Concretely, a low score indicates that the verification method is accurate – i.e the over approximation is closer to the true value. We observe that the tighter the bounds, the higher the scores. This allows us to conjecture that the more scalable the formal methods will become, the better the quality of the generated explanations will be.

5. Conclusion

In this work, we presented the first explainability method that uses verification perturbation analysis to have an exhaustive exploration of a perturbation space to generate explanations. We presented an efficient estimator that yields explanations which are state-of-the-art on current metrics.

We also described a simple strategy to scale up the approach in anticipation of improvements in perturbation verification methods. Finally, we showed that this estimator can be used to form easily interpretable targeteted explanations.

We hope that this work will help guide further developments – searching for safer and more efficient explanation methods for neural networks – and that it will inspire further synergies with the field of formal verification.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. 7, 15

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 1

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for deep neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 13

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010. 2

Balunovic, M. and Vechev, M. Adversarial training and provable defenses: Bridging the gap. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 5

Bhatt, U., Weller, A., and Moura, J. M. F. Evaluating and aggregating feature-based model explanations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2020. 14

Boopathy, A., Liu, S., Zhang, G., Liu, C., Chen, P.-Y., Chang, S., and Daniel, L. Proper network interpretability helps adversarial robustness in classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. 2

Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2018. 7, 14

Chollet, F. et al. Keras. https://keras.io, 2015. 7

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 5, 6

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *ArXiv e-print*, 2017. 1

Ducoffe, Melanie. Decomon: Automatic certified perturbation analysis of neural networks, 2021. URL https://github.com/airbus/decomon. 3

Ehlers, R. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pp. 269–286. Springer, 2017. 3

Fel, T. and Vigouroux, D. Representativity and consistency measures for deep neural network explanations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022. 2

Fel, T., Cadene, R., Chalvidal, M., Cord, M., Vigouroux, D., and Serre, T. Look at the variance! efficient black-box explanations with sobol-based sensitivity analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 4, 5

Fel, T., Hervier, L., Vigouroux, D., Poche, A., Plakoo, J., Cadene, R., Chalvidal, M., Colin, J., Boissin, T., Béthune, L., Picard, A., Nicodeme, C., Gardes, L., Flandin, G., and Serre, T. Xplique: A deep learning explainability toolbox. *Workshop, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 13

Ferrettini, G., Escriva, E., Aligon, J., Excoffier, J.-B., and Soulé-Dupuy, C. Coalitional strategies for efficient individual prediction explanation. *Information Systems Frontiers*, pp. 1–27, 2021. 5

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 5

Ghalebikesabi, S., Ter-Minassian, L., DiazOrdaz, K., and Holmes, C. C. On locality of local explanation models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

Ghorbani, A., Abid, A., and Zou, J. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2017. 1

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 13

Hsieh, C.-Y., Yeh, C.-K., Liu, X., Ravikumar, P., Kim, S., Kumar, S., and Hsieh, C.-J. Evaluations and methods for explanation through robustness analysis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 2, 3, 4, 5, 7, 12, 14

Ignatiev, A., Narodytska, N., and Marques-Silva, J. Abduction-based explanations for machine learning models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019a. 2, 5, 12

Ignatiev, A., Narodytska, N., and Marques-Silva, J. On relating explanations and adversarial examples. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019b. 2, 12

Kaminski, M. E. The right to explanation, explained. In *Research Handbook on Information Law and Governance*. Edward Elgar Publishing, 2021. 8

Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pp. 97–117. Springer, 2017. 3

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images, 2009. 6

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105, 2012. 3

LeCun, Y. and Cortes, C. MNIST handwritten digit database, 2010. 6

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 2015. 1

Lin, Z. Q., Shafiee, M. J., Bochkarev, S., Jules, M. S., Wang, X. Y., and Wong, A. Do explanations reflect decisions? a machine-centric strategy to quantify the performance of explainability algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 2, 3

Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 1, 4

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 7, 14

O'Shaughnessy, M., Canal, G., Connor, M., Davenport, M., and Rozell, C. Generative causal explanations of black-box classifiers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. 1, 2, 3, 4, 5, 7, 14

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. In *Knowledge Discovery and Data Mining (KDD)*, 2016. 1, 2

Ross, A., Lakkaraju, H., and Bastani, O. Learning models for actionable recourse. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

Salman, H., Yang, G., Zhang, H., Hsieh, C.-J., and Zhang, P. A convex relaxation barrier to tight robustness verification of neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 2016. 7

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 13

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017. 1, 13

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop, Proceedings of the International Conference on Learning Representations (ICLR)*, 2013. 1, 13

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 2, 15

Singh, G., Gehr, T., Püschel, M., and Vechev, M. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 2019. 3

Slack, D., Hilgard, A., Lakkaraju, H., and Singh, S. Counterfactual explanations can be manipulated. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021a. 1

Slack, D., Hilgard, A., Singh, S., and Lakkaraju, H. Reliable post hoc explanations: Modeling uncertainty in explainability. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021b. 2

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. In *Workshop on Visualization for Deep Learning, Proceedings of the International Conference on Machine Learning (ICML)*, 2017. 1, 2, 13

Sotoudeh, M. and Thakur, A. V. Computing linear restrictions of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 13

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *Workshop Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 2

Sturmfels, P., Lundberg, S., and Lee, S.-I. Visualizing the impact of feature attribution baselines. *Distill*, 2020. 2, 7

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017. 1, 2, 13

Tjeng, V. and Tedrake, R. Verifying neural networks with mixed integer programming. *Proceedings of the International Conference on Learning Representations (ICLR)*, 15, 2019. 3

Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.-J., and Kolter, J. Z. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

Xu, K., Shi, Z., Zhang, H., Wang, Y., Chang, K.-W., Huang, M., Kailkhura, B., Lin, X., and Hsieh, C.-J. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2014a. 1, 14

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2014b. 2

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

A. EVA and *Robustness*-S_r

We show here that the explanations generated by EVA provides an optimal solution from a certain stage to the *Robustness*-S_r metric proposed by (Hsieh et al., 2021). 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 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 A.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 variable to guarantee a decision is also a standard research problem (Ignatiev et al., 2019a;b) and is called subset-minimal explanations.

Intuitively, the optimal set is the combination of variables that allows to find 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 an 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 essential variable before showing the optimality of $\boldsymbol{\delta}^*$.

Definition A.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 allow us to characterize the optimal set at step k.

Proposition A.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 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 A.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 A.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 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 A.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 A.4 and Proposition A.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 A.3, $\boldsymbol{u}$ is the optimal set for k, with $k \in [\![|\boldsymbol{u}|, d]\!]$. □

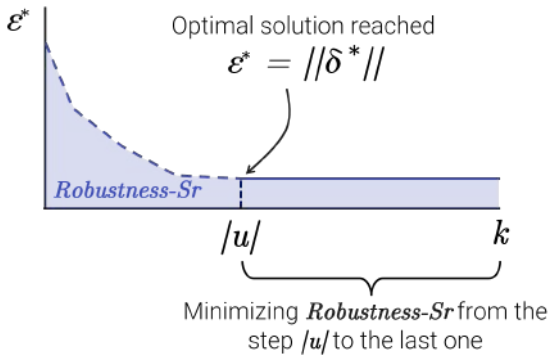

Figure 6. **EVA yield optimal subset of variable from step $|\boldsymbol{u}|$.** $Robustness\text{-}S_r$ measures the AUC of the distances to the nearest adversary for the k most important variables. With $\boldsymbol{\delta}^*$ the nearest reachable adversarial perturbation around $\boldsymbol{x}$, 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.

B. 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 (Fel et al., 2022). 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 variables x_i. We will denote the explanation functionnal mapping an input of interest $\boldsymbol{x} = (x_1, ..., x_d) \in \mathcal{X}$ as $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^d$. All the attribution

[fel]: finish ? error ?

Saliency (Simonyan et al., 2013) 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.

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

Gradient $\odot$ Input (Shrikumar et al., 2017) 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 (Ancona et al., 2018) showed that Gradient $\odot$ Input is equivalent to ϵ-LRP and DeepLIFT (Shrikumar et al., 2017) 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 (Sundararajan et al., 2017) 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 (Sotoudeh & Thakur, 2019) 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 (Smilkov et al., 2017) 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.

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

VarGrad (Hooker et al., 2019) is similar to SmoothGrad as it employs the same methodology to construct the attribution maps: using a set of m noisy inputs, it aggregate the gradients using the variance rather than the mean. For the experiment, m and σ are the same as Smoothgrad. Formally:

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

Grad-CAM (Selvaraju et al., 2017) 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 experiment, we used the last convolutional layer of each model to compute the explanation.

Grad-CAM++ (G+) (Chattopadhay et al., 2018) 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 (Zeiler & Fergus, 2014a) is a sensitivity method that sweep 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 (Petsiuk et al., 2018) is a black-box method that consist 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 (Hsieh et al., 2021) is a greedy like method which aggregate 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 $b : \mathcal{P}(\overline{\boldsymbol{u}}) \to \{0,1\}^{|\overline{\boldsymbol{u}}|}$ 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 solve the following regression problem:

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

The weights obtained indicate the importance of each variable by taking into account these interactions. We specify that $v(.)$ 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 (Madry et al., 2018) 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 get a better score.

C. 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 softmax of the models.

Deletion. (Petsiuk et al., 2018) The first metric is Deletion, it consists in measuring the drop of 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 (Hsieh et al., 2021), the baseline state is not fixed but is a value drawn on a uniform distribution $\boldsymbol{x}_0 \sim \mathcal{U}(0,1)$.

Insertion. (Petsiuk et al., 2018) 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.

μFidelity (Bhatt et al., 2020) 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:

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

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.

D. Models

The models used were all trained using Tensorflow (Abadi et al., 2015). For Mnist, the model is a stacking of 5 Dense layer 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 achieve 75% of accuracy on the test set. For ImageNet, we used a pre-trained VGG16 (Simonyan et al., 2014).

E. 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 4.2.

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, a high importance for $\boldsymbol{\phi}_{\boldsymbol{u}}^{(+)}$ means that the model is sensitive to the addition of a white line. Conversely, a 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 Figure 7.

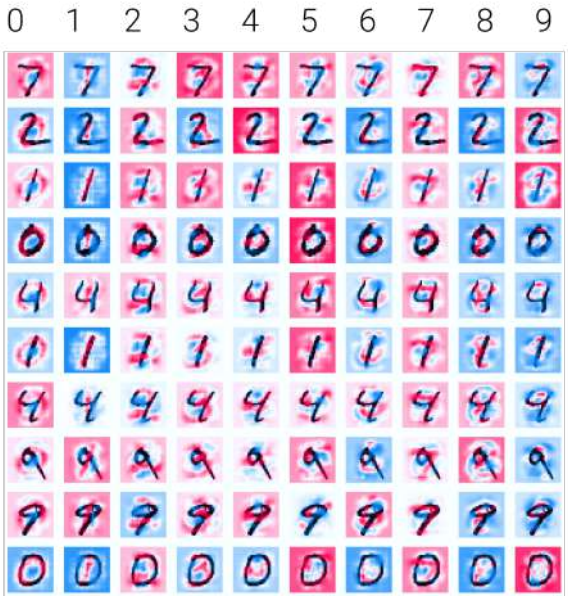

Figure 7. **Targeted Explanations** Attributions generated explanation for a decision other than the one predicted. Each column represent the class explained – e.g, the first column look for explanation for the class '0' for each of samples. As indicated in section 4.2, the red areas indicate that a black line should be added and the blue areas that it should be removed. More examples are available in the appendix.