



Evaluating feature attribution methods in the image domain

Arne Gevaert¹ · Axel-Jan Rousseau² · Thijs Becker⁴ · Dirk Valkenborg² · Tijl De Bie³ · Yvan Saeys¹

Received: 25 May 2022 / Revised: 13 April 2024 / Accepted: 17 April 2024 /

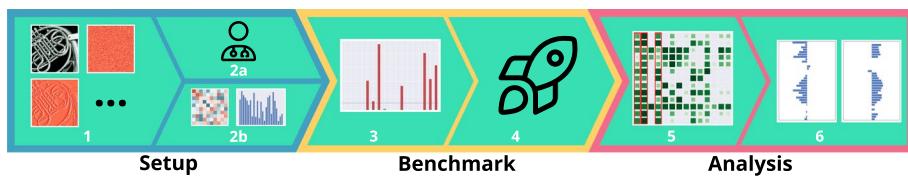
Published online: 24 May 2024

© The Author(s) 2024

Abstract

Feature attribution maps are a popular approach to highlight the most important pixels in an image for a given prediction of a model. Despite a recent growth in popularity and available methods, the objective evaluation of such attribution maps remains an open problem. Building on previous work in this domain, we investigate existing quality metrics and propose new variants of metrics for the evaluation of attribution maps. We confirm a recent finding that different quality metrics seem to measure different underlying properties of attribution maps, and extend this finding to a larger selection of attribution methods, quality metrics, and datasets. We also find that metric results on one dataset do not necessarily generalize to other datasets, and methods with desirable theoretical properties do not necessarily outperform computationally cheaper alternatives in practice. Based on these findings, we propose a general benchmarking approach to help guide the selection of attribution methods for a given use case. Implementations of attribution metrics and our experiments are available online (<https://github.com/arnegevaert/benchmark-general-imaging>).

Graphical abstract



Keywords Explainability · Interpretability · Benchmark · Feature attribution · Saliency maps

Editor: Johannes Fürnkranz.

Extended author information available on the last page of the article

1 Introduction

Deep neural networks have for some years been the state of the art for a number of predictive tasks, such as image classification (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; He et al., 2016), language modeling (Vaswani et al., 2017; Brown et al., 2020) and reinforcement learning (Mnih et al., 2015; Silver et al., 2017; Lillicrap et al., 2019). This has led to their widespread adoption in many areas of machine learning. Such models are however notorious for their black box nature: due to the large numbers of parameters and complex neural architectures, their predictions become very difficult or even impossible to understand. Interpretability of predictive models is a very useful property for many different reasons: it allows us to extract understandable knowledge from large datasets, potentially leading to new knowledge about the data itself, debug models when they fail, and explain predictions to end users to build trust in the system (Doshi-Velez & Kim, 2017). In some cases, the ability to explain predictions is crucial for model deployment.

For this reason, a number of different techniques have been proposed to try to make neural networks more explainable. Proposed approaches include extracting interpretable rules (Ribeiro et al., 2018), counterfactual explanations (Wachter et al., 2017; Dandl et al., 2020), model distillation (Liu et al., 2018), and feature attribution (Ribeiro et al., 2016; Selvaraju et al., 2017; Sundararajan et al., 2017; Springenberg et al., 2014; Simonyan et al., 2014; Erion et al., 2021; Smilkov et al., 2017). In this work, we focus on the latter type of explanation. Feature attribution explanations are among the most popular techniques for explaining image classification models, because they can easily be visualized as a heatmap showing which pixels in an image are important (in the case of color images, the attribution value of a pixel can be defined to be the average or maximum absolute value of its three color components). Feature attribution techniques can also be used to measure the importance of hidden neurons or layers (Shrikumar et al., 2017; Selvaraju et al., 2017), although the focus in this work is on pixel attribution in the image domain.

The exact task of feature attribution can be interpreted in different ways, leading to some discussion about which properties a feature attribution method should satisfy (Chen et al., 2020). Feature attribution methods can roughly be divided in four categories using two properties: local vs. global, and model- vs. data-centric. The first property concerns the scope of the explanation: local feature attribution maps the importance of features *in a given sample*, whereas global feature attribution maps the importance of features for *all samples* in a dataset (also called feature *importance*). The second property concerns the target of the explanation: model-centric feature attribution concerns the importance of features *for a specific model*, whereas data-centric feature attribution measures the *informativeness* of features in the data, independently of any specific model. This can be estimated using classical statistical or information-theoretical techniques (Chandrashekhar & Sahin, 2014). Model- and data-centric feature attributions are not necessarily the same, as a model can often make predictions using only a subset of the informative features, or even using features that are generally non-informative (in which case the model is overfitting). In this work, we specifically evaluate local, model-centric feature attributions.

Because of the desire for model explanations and the widespread popularity of deep neural networks in the domain of image classification, many feature attribution methods have been proposed in recent years. These methods can roughly be divided into gradient-based, perturbation-based and CAM-based methods (Ancona et al., 2018; Selvaraju et al., 2017). Each attribution method creates a different explanation for the same prediction. This has naturally led to the question of explanation quality: which methods work best? This

turns out to be a very difficult question, since there is no ground truth available in the form of “perfect” feature attribution scores.

Attempts at evaluating feature attribution explanations can roughly be categorized in three types of approaches. The first is human evaluation. This includes simply looking at an explanation and seeing if it “makes sense”, or performing a user study to see how helpful explanations are for predicting model behaviour (Schmidt & Biessmann, 2019). The disadvantage of these approaches is that such user studies are difficult to set up, and their results are inherently subjective. It has been shown that, just because an explanation makes sense to humans, does not mean that it is true to the underlying workings of the model (Adebayo et al., 2018). Also, a user study is generally infeasible to perform for each use case, and it is unclear whether results from user studies can be generalized to different datasets.

A second approach is to define a set of desirable properties, or *axioms*, that a method should have (Lundberg & Lee, 2017; Sundararajan et al., 2017). Examples of such axioms include local accuracy, missingness and consistency (Lundberg & Lee, 2017). Such approaches are more objective in nature, but recent work has shown that methods that conform to these axioms are still not necessarily accurate (Adebayo et al., 2018). Some of these axioms can also be implemented in different ways, leading to a number of methods that all conform to certain axioms, but still provide different explanations for the same prediction (Sundararajan & Najmi, 2020).

Finally, we can define quantitative metrics that try to indicate the quality of an explanation by measuring the behaviour of the model or explanation after applying specific perturbations (Ancona et al., 2018; Yeh et al., 2019). A simple example of this kind of measure is Deletion (Samek et al., 2017). Here, we iteratively mask the top n most important features, as indicated by the explanation. If the features that were marked as important are truly important, we would expect the output of the model to drop rapidly with increasing n . Another example is the so-called *sanity check* proposed by Adebayo et al. (2018). This sanity check works by randomizing the parameters of the model, and comparing the original attribution map to the attribution map computed for the randomized model. If these two maps are similar, then the attribution method is independent of the model parameters, and is therefore viewed as *passing* the sanity check.

In this work, we implement several existing and newly proposed quality metrics for evaluating feature attribution methods. These metrics are evaluated on a large number of attribution methods, and we investigate the results on 8 different datasets of varying dimensionality. Our contributions are as follows:

- We expand on the work done in Tomsett et al. (2020), showing that different quality metrics measure different underlying properties of attribution maps. We extend this finding to a significantly larger set of quality metrics, attribution methods, and datasets.
- We demonstrate that the results of quality metrics for attribution maps, including the sanity check from Adebayo et al. (2018), vary significantly across different datasets. From this observation, we conclude that quality metrics should be computed separately for each given use case, rather than assuming that the results for one dataset or model will generalize to another setting.
- We propose three new metrics: Minimal Subset Deletion, Minimal Subset Insertion and Seg-Sensitivity- n (based on Sensitivity- n (Ancona et al., 2018)). We empirically show that Seg-Sensitivity- n provides results with a higher signal-to-noise ratio on high-dimensional datasets.
- We find that the performance of some methods is complementary to that of other methods, suggesting that a combination of these attribution methods may be valuable.

- We find that, depending on the dataset, methods with strong theoretical foundations such as DeepSHAP (Lundberg & Lee, 2017) do not necessarily outperform their computationally cheaper counterparts such as DeepLIFT (Shrikumar et al., 2017). This suggests that a benchmarking approach can be useful to check if a computationally intensive method is truly more valuable than a simpler one for a given use case.
- Finally, we provide general benchmarking guidelines to help guide the search for an appropriate attribution method or set of attribution methods for a given use case.

2 Related work

Although the systematic evaluation of feature attribution methods is a fairly recent topic, a number of attempts have already been made to systematically and objectively evaluate the quality of explanations. An early, intuitive way of evaluating feature attributions was proposed by Samek et al. (2017). In this approach, the top k most important features are removed by replacing them with random noise. Consequently, the difference in output of the model is measured. If the most important features are truly important to the model, we expect a sharp drop in confidence for the predicted class.

A more general approach was proposed by Ancona et al. (2018), called Sensitivity- n . This metric is computed by removing a number of random subsets of n pixels from the image, and measuring the correlation of the difference in output with the sum of attribution values of those removed pixels. This allows one to assess the accuracy of the attribution values *in general*, rather than just the top most important features.

A possible problem with the metrics mentioned above is the fact that masking inputs in images can introduce high-frequency artifacts, which can push the images outside of their normal data distribution. This can cause the model to produce arbitrary outputs. Although the exact impact of this problem on the scores produced by metrics is unclear, some efforts to resolving it have already been made, including the Remove And Retrain (ROAR) procedure (Hooker et al., 2019). Here, the authors attempt to resolve the OOD problem by modifying the Deletion metric by Samek et al. (2017) such that after every iteration, the model is retrained on the data where the top k pixels are removed. The reasoning is that in this way, the model learns to regard the mean-valued pixels as uninformative.

However, we argue that this metric is not measuring the same kind of feature attribution as the original Deletion metric. Because the model is retrained after each iteration, it is able to detect and use different parts of the input to make a prediction. Also, there is no guarantee that the model, after retraining, will consider the masked pixels as uninformative: the shape or location of regions with that specific color (the dataset mean) can still be very informative. In other words, ROAR can only assess the ability of methods to map local, *data-centric* feature attributions. Since the methods we are evaluating are designed to map local, model-centric attributions, we do not incorporate this metric in our analysis.

More recently, Yeh et al. (2019) proposed Infidelity and Max-Sensitivity, two complementary metrics that measure the accuracy of a method and its robustness against small, insignificant perturbations, respectively. Recent work has shown that some feature attribution methods, much like neural networks themselves, are vulnerable to adversarial attacks (Ghorbani et al., 2019). This makes the robustness of explanations an interesting property to measure in addition to the accuracy.

Yang and Kim (2019) proposed a synthetic data approach, where objects from MSCOCO (Lin et al., 2014) were pasted into background images from MiniPlaces (Zhou

et al., 2017). A model is then trained to classify either the background or the object in the image. Because it is known where in the image the object was pasted, a relative form of ground truth is available. From this, a number of metrics are derived. However, as opposed to the previously proposed metrics, these metrics are tied to a specific dataset, and cannot be calculated for any given dataset and model. For this reason, we do not consider this approach in our work.

Another related approach was proposed by Adebayo et al. (2018). In this work, a relative form of ground truth is created by randomizing the parameter values of the network, layer by layer. The assumption is that the feature attribution map should be significantly different for a trained model vs. a randomized model. Methods that return the same attribution map for both models, appear to be independent of the model parameters. This approach does not provide a numerical value that captures the quality of the explanation, but rather acts as a pass/fail “sanity check”.

Recently, Tomsett et al. (2020) have shown that some of these metrics are very dependent on implementation details, and do not appear to be measuring the same underlying properties of explanations. This is shown by measuring the correlation between different metric scores. The authors find that details such as how pixels are masked (by setting them to 0 vs. replacing them with random noise), or in what order they are masked (by decreasing or increasing importance), have a great influence on the quality scores given by the metric. This suggests that these metrics, although they are all designed to measure the “accuracy” of explanations, appear to be measuring different underlying properties. We build upon this work by applying a similar but more extensive analysis on a larger number of metrics and methods. In doing so, we can draw more global conclusions about how different methods and metrics relate to each other, and which methods and metrics may be most desirable for specific use cases.

3 Definitions and notation

We define an *instance* as a vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, where d is the number of inputs (pixels, or color values of pixels in the case of RGB images). A *model* is defined as a function $m : \mathcal{X} \rightarrow \mathbb{R}^o$, where o is the number of output classes. Note that the output of the model for a given class c can be any real number. In many cases, the output of the model is followed by a softmax function σ , mapping the outputs from \mathbb{R}^o to $[0, 1]^o$. In this case, the original outputs are called *logits*. In this work, we consider the logit values as the actual output of the model. We write $m(\mathbf{x})_c$ as the c -th component of the output of model m on instance \mathbf{x} .

We denote the *model class* as $\mathcal{M} = \{m\}$, this represents the set of possible model instantiations (for example, the set of all possible neural networks, or all possible neural networks of a given architecture). An *attribution method* is a function $E \in \mathcal{E} : \mathcal{M} \times \mathcal{X} \times \{1, \dots, o\} \rightarrow \mathbb{R}^d$. The result of this function is called an *attribution map*. We explicitly mention the model $m \in \mathcal{M}$ as an argument of this function to indicate that we consider local, model-centric attributions, which are dependent on a specific combination of instance and model. The output class is also an argument of the attribution method, as attributions can be calculated for each output of the model. Finally, we define an *attribution metric* as a function $M : \mathcal{M} \times \mathcal{X} \times \mathbb{R}^d \times \{1, \dots, o\} \rightarrow \mathbb{R}$ (type I) or $M : \mathcal{M} \times \mathcal{X} \times \mathcal{E} \times \{1, \dots, o\} \rightarrow \mathbb{R}$ (type II), mapping a model m , an instance \mathbf{x} , an attribution map \mathbf{e} (type I) or attribution method E (type II), and an output class c to a single

Table 1 Summary of metrics

Metric	Range	Masking	Data type	Complexity	Interface
Del_{MoRF}	High-end	✓	Any	$\mathcal{O}(L)$	Type I
Del_{LeRF}	Low-end	✓	Any	$\mathcal{O}(L)$	Type I
Ins_{MoRF}	High-end	✓	Any	$\mathcal{O}(L)$	Type I
Ins_{LeRF}	Low-end	✓	Any	$\mathcal{O}(L)$	Type I
MS_{Del}	High-end	✓	Any	$\mathcal{O}(d)$	Type I
MS_{Ins}	High-end	✓	Any	$\mathcal{O}(d)$	Type I
$IROF_{MoRF}$	High-end	✓	Image	$\mathcal{O}(S)$	Type I
$IROF_{LeRF}$	Low-end	✓	Image	$\mathcal{O}(S)$	Type I
$Sens_n$	Overall	✓	Any	$\mathcal{O}(k)$	Type I
$SegSens_n$	Overall	✓	Image	$\mathcal{O}(k)$	Type I
$INFD$	Overall	✓	Any	$\mathcal{O}(k)$	Type I
$SENS_{MAX}$	Overall		Any	$\mathcal{O}(k * C_{mth})$	Type II
COV	High-end		Image	C_{mth}^1	Type II
PR	Overall		Any	C_{mth}	Type II

¹provided that an adversarial patch is already available

real number which represents the quality of the attributions given by \mathbf{e} or E for output c of model m on instance \mathbf{x} .

For an instance \mathbf{x} and attribution map \mathbf{e} , we will denote $\mathbf{x}_k^{\mathbf{e}}$ and $\mathbf{x}_{-k}^{\mathbf{e}}$ as the instance \mathbf{x} where respectively the k most or least important inputs are removed according to \mathbf{e} . This removal, or “masking out” of features can be implemented in a number of different ways, which will be discussed in detail in Sect. 6.2. In the case of color images, we define the attribution value of a pixel as the average value of its color components, and proceed analogously.

Finally, we will denote $S = \{S^l\}_{l=1}^L$, $S^l \in \{0, 1\}^d$ as the set of segments of an input sample \mathbf{x} as produced by a given segmentation algorithm, where $S_i^l = 1$ if the i th input feature is part of segment l , and $S_i^l = 0$ otherwise. The attribution value of a segment S^l can then simply be computed as the average attribution value of its input features: $\mathbf{e}_{S^l} := \frac{\|\mathbf{e} \odot S^l\|_1}{\|S^l\|_1}$, where \odot indicates element-wise multiplication. For an instance \mathbf{x} , a corresponding segmentation S , and an attribution map \mathbf{e} , we will denote $\mathbf{x}_{k_s}^{\mathbf{e}}$ (resp. $\mathbf{x}_{-k_s}^{\mathbf{e}}$) as the same sample \mathbf{x} where the k most (resp. least) important segments are masked out.

4 Attribution metrics

We now describe the different quality metrics that were used to evaluate the attribution methods described in Sect. 5. A summary of general properties can be seen in Table 1:

- *Attribution range* Indicates which parts of the attribution map the metric actually evaluates: we denote metrics that evaluate the most important, least important, or all inputs as *high-end*, *low-end* or *overall* metrics, respectively. For example: Del_{MoRF} and Del_{LeRF} evaluate the high- and low-end, respectively, because they measure the influence of removing the top and bottom k features, respectively (see further).

- *Masking* Indicates whether the metric relies on masking inputs in its implementation. Metrics that do, can be implemented in different ways, as the choice of a neutral value to replace features with is not obvious (see Sect. 6.2).
- *Data type* Indicates which types of data the metric can be applied to. In our case, a metric can either be applied to any kind of data, or only to image data (for example, because it relies on an image segmentation algorithm (Rieger & Hansen, 2020), or an adversarial patch (Lin et al., 2019)).
- *Complexity* Indicates the computational complexity of the metric expressed as a number of forward passes through the model. C_{mth} is the complexity of the attribution method being evaluated, also expressed as a number of forward/backward passes.
- *Interface* We define two interfaces for attribution metrics:
 - *Type I* $M : \mathcal{M} \times \mathcal{X} \times \mathbb{R}^d \times \{1, \dots, o\} \rightarrow \mathbb{R}$. A type I metric accepts an *attribution map* $\mathbf{e} \in \mathbb{R}^d$ to evaluate. This allows one to compute the metric result for any attribution map, regardless of whether the implementation of the attribution method that generated it is available or not.
 - *Type II*: $M : \mathcal{M} \times \mathcal{X} \times \mathcal{E} \times \{1, \dots, o\} \rightarrow \mathbb{R}$. A type II metric needs access to the *attribution method* under evaluation $E \in \mathcal{E}$, because the method needs to be re-executed at some point in the computation of the metric. If the implementation of the attribution method is not available, this type of metric cannot be computed.

4.1 Deletion

The first and most widely known metric is Deletion (Samek et al., 2017). This metric works by iteratively removing the top k most important features from an input sample. This is done by masking the feature with some value (see further: 6.2).

An ordering of features where the most important features are ranked highest will cause a steep decrease in the output confidence of the model. This can be summarized by computing the area under the perturbation curve, where a low AUC corresponds to a good explanation. Samek et al. (2017) also introduces an alternative variant of the Deletion metric, where the features are masked in reversed order of importance. In that case, a high AUC value indicates a good attribution map. We call this variant $\text{Deletion}_{\text{LeRF}}$ (Least Relevant First), and the original $\text{Deletion}_{\text{MoRF}}$ (Most Relevant First).

$$\text{Del}_{\text{MoRF}}(\mathbf{x}, m, \mathbf{e}, c) = \frac{1}{L} \sum_{k=1}^L m(\mathbf{x}_k^{\mathbf{e}})_c$$

$$\text{Del}_{\text{LeRF}}(\mathbf{x}, m, \mathbf{e}, c) = \frac{1}{L} \sum_{k=1}^L m(\mathbf{x}_{-k}^{\mathbf{e}})_c$$

Where L is the maximum number of inputs masked, and c is the output that the attribution is intended to explain (usually this is the highest output of the model, which corresponds to the class that the model assigned to \mathbf{x}). For large images, we can approximate this value by taking a fixed number of steps with a constant step size. The MoRF-variant evaluates the high end of the attribution map, whereas the LeRF-variant evaluates the low end. We choose L such that at most 15% of pixels are masked, which corresponds to the original approach in Samek et al. (2017). This limits the influence of out-of-distribution effects: as

more pixels are removed, the image gets further removed from the original data manifold, making the result less representative. This metric scales linearly in the number of steps L taken to compute the AUC.

4.2 Insertion

A simple variant of the Deletion metric is Insertion (Petsiuk et al., 2018). This metric works entirely analogously to Deletion, but instead of iteratively removing features from the original input sample, we now iteratively insert pixels of the original image onto a blank background (which is again defined by the masking procedure).

Analogously to the Deletion metric, we can again define two variants of Insertion: Ins_{LeRF} and Ins_{MoRF} , where resp. the least and most relevant features are inserted first. Since inserting the k most important features is the same as removing the $d - k$ least important ones, we can define Insertion as follows:

$$Ins_{MoRF}(\mathbf{x}, m, \mathbf{e}, c) = \frac{1}{L} \sum_{k=1}^L m(\mathbf{x}_{-(d-k)}^{\mathbf{e}})_c$$

$$Ins_{LeRF}(\mathbf{x}, m, \mathbf{e}, c) = \frac{1}{L} \sum_{k=1}^L m(\mathbf{x}_{d-k}^{\mathbf{e}})_c$$

Note that if $L = d$, $Ins_{MoRF} = Del_{LeRF}$ and $Ins_{LeRF} = Del_{MoRF}$. Again, the MoRF- and LeRF-variants measure the high and low end respectively. This metric also scales linearly with the number of steps hyperparameter L .

4.3 Minimal subset

The previously mentioned Deletion and Insertion metrics only take into account the model's confidence in the originally predicted class c . However, the actual prediction of the model is also dependent on the confidence of the other classes. The removal of certain features could, for example, hardly influence the output confidence in c , but drastically change the confidence of another class c' , causing the model to change its overall prediction. To mitigate this problem, we introduce Minimal Subset Deletion and Minimal Subset Insertion.

These metrics work by iteratively removing (resp. inserting) the top k most important features from the input sample, and recording the smallest value for k that causes the prediction of the model to change. For Minimal Subset Insertion specifically, the prediction must change into the originally predicted class c .

$$MS_{Del}(\mathbf{x}, m, \mathbf{e}, c) = \arg \min_{k \in \{1, \dots, d\}} (\operatorname{argmax}(m(\mathbf{x}_k^{\mathbf{e}})) \neq \operatorname{argmax}(m(\mathbf{x})))$$

$$MS_{Ins}(\mathbf{x}, m, \mathbf{e}, c) = \arg \min_{k \in \{1, \dots, d\}} (\operatorname{argmax}(m(\mathbf{x}_{-(d-k)}^{\mathbf{e}})) = \operatorname{argmax}(m(\mathbf{x})))$$

For analogous reasons as with Deletion/Insertion, this metric evaluates the high end of the attribution map. Both variants scale linearly with the amount of dimensions d .

4.4 IROF

Iterative Removal Of Features (IROF) (Rieger & Hansen, 2020) is an extension of Deletion, where a segmentation S of the input sample \mathbf{x} is used. Instead of iteratively masking the k most important inputs, we now mask the k most important *segments*. This can reduce the number of forward passes needed, and can provide insight into the quality of an attribution at a larger scale: an algorithm that is able to find the top few pixels that maximally perturb the network when removed, might score very well on Deletion, but not so much on IROF. If another algorithm correctly identifies the most important “regions”, it might score better on IROF and worse on Deletion. In some cases, the latter might be more interesting, as this would likely correspond to explanations that are less noisy and more easily readable.

We can define $IROF_{MoRF/LeRF}$ analogously to $Deletion_{MoRF/LeRF}$, leading to the following definitions:

$$IROF_{MoRF}(\mathbf{x}, m, \mathbf{e}, c) = \frac{1}{|S|} \sum_{k=1}^{|S|} m(\mathbf{x}_{k_s}^{\mathbf{e}})_c$$

$$IROF_{LeRF}(\mathbf{x}, m, \mathbf{e}, c) = \frac{1}{|S|} \sum_{k=1}^{|S|} m(\mathbf{x}_{-k_s}^{\mathbf{e}})_c$$

Note that, even though all segments are removed in IROF, we classify this metric as high-end. This is because the metric score still depends most on the top most important image segments: if those are identified correctly, the model output will decrease quickly, and the other segments will have little influence on the metric score. IROF scales linearly with the amount of segments $|S|$, and is only applicable to image data because of the dependence on an image segmentation algorithm. We implement IROF using the SLIC algorithm (Achanta et al., 2012), with an approximate number of segments of 100.

4.5 Sensitivity-n

Previous metrics have only considered the most or least important features. This can be a problem if the inputs contain a large number of features, in which case a large proportion of the features is hardly evaluated, or has a small influence on the evaluation. To get a more global assessment of the quality of feature attributions, Sensitivity-n was introduced (Ancona et al., 2018). Formally, Sensitivity-n is defined as follows (quoted from Ancona et al., 2018, where mathematical notation was adjusted to conform to ours):

An attribution method satisfies Sensitivity- n when the sum of the attributions for any subset of features of cardinality n is equal to the variation of the output $m(\mathbf{x})_c$ caused by removing the features in the subset.

Since no attribution method exactly satisfies Sensitivity- n for all values of n , the metric instead measures how well the sum of attributions $\sum_{s \in S} \mathbf{e}_s$ correlates with the difference in output $m(\mathbf{x})_c - m(\mathbf{x}_S)_c$, using the Pearson correlation coefficient (where \mathbf{x}_S denotes the instance \mathbf{x} with all features in S removed, and \mathbf{e}_s denotes the attribution of feature s according to attribution map \mathbf{e}). We can compute Sensitivity- n as:

$$Sens_n(\mathbf{x}, m, \mathbf{e}, c) = r \left(\sum_{s \in S_i} \mathbf{e}_s, m(\mathbf{x})_c - m(\mathbf{x}_{S_i})_c \right)$$

Where S_i is a random subset of inputs of size n , and $r(X, Y)$ is the Pearson correlation coefficient between variables X and Y . The correlation is computed using k randomly selected subsets S_i . We choose $k = 100$, which corresponds to the configuration in Ancona et al. (2018).

The number of possible subsets of features grows exponentially with d . Because of this, the approximation made by this metric will get exponentially worse for increasing image size. To mitigate this problem, we introduce a segmented variant of Sensitivity-n, called *Seg-Sensitivity-n*. This metric works by first segmenting the input image \mathbf{x} into segments S , and then removing random subsets of segments instead of features. Since the amount of segments is drastically lower than the number of features, selecting 100 random subsets gives a more representative sample, which we expect will increase the signal-to-noise ratio of this metric.

$$SegSens_n(\mathbf{x}, m, \mathbf{e}, c, S) = r \left(\sum_{l \in L_i} \mathbf{e}_{S^l}, m(\mathbf{x})_c - m(\mathbf{x}_{S^{L_i}})_c \right)$$

Where S is the segmentation of instance \mathbf{x} (represented as a set of segments $\{S^l\}$), and $\mathbf{x}_{S^{L_i}}$ denotes the instance \mathbf{x} with all segments in L_i removed. The correlation is now computed using $k = 100$ randomly selected subsets of segments L_i . Since the subsets of features/segments are chosen randomly, Sensitivity-n and Seg-Sensitivity-n evaluate the overall attribution map. Both metrics scale linearly in the number of subsets k .

4.6 Infidelity

Infidelity (Yeh et al., 2019) generalizes the previous metrics from perturbation by masking to general perturbations. This is done by comparing the difference in output after an arbitrary perturbation with the dot product of the perturbation vector \mathbf{I} and the attribution map \mathbf{e} . The perturbation vector is a random variable $\mathbf{I} \in \mathbb{R}^d$ with probability measure $\mu_{\mathbf{I}}$. The infidelity of an attribution map \mathbf{e} for an input sample \mathbf{x} and class c is then defined as follows:

$$INFD(\mathbf{x}, m, \mathbf{e}, c) = \mathbb{E}_{\mathbf{I} \sim \mu_{\mathbf{I}}} [(\beta \mathbf{I}^T \mathbf{e} - (m(\mathbf{x})_c - m(\mathbf{x} - \mathbf{I})_c))^2]$$

$$\beta = \frac{\mathbb{E}_{\mathbf{I} \sim \mu_{\mathbf{I}}} [\mathbf{I}^T \mathbf{e} (m(\mathbf{x})_c - m(\mathbf{x} - \mathbf{I})_c)]}{\mathbb{E}_{\mathbf{I} \sim \mu_{\mathbf{I}}} [(\mathbf{I}^T \mathbf{e})^2]}$$

Here, β acts as a normalizing term (called *optimal scaling* in the original paper) to make the values for different explanation methods comparable. We use two variants of Infidelity proposed in Yeh et al. (2019), defined by their perturbation vectors:

- Difference to noisy baseline ($INFD_{NB}$): $\mathbf{I} = \mathbf{x} - \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. This corresponds to a robust variant of the completeness axiom (Lundberg & Lee, 2017), where we take a Gaussian random vector centered around a zero baseline, instead of a constant zero baseline.

- Square removal (INF_{SQ}): in this case, \mathbf{I} has a uniform distribution over square patches of the image \mathbf{x} of some predefined size. This can better capture spatial relationships in the images, as the removal of single pixels actually removes very little information if the surrounding pixels are still intact.

Since the perturbations happen on the entire image or on randomly selected squares, respectively, they evaluate the overall attribution map. Infidelity scales linearly with the number of samples k used to approximate the expected value. In this work, $k = 1000$, which corresponds to the original implementation by Yeh et al. (2019).

4.7 Max-sensitivity

Max-Sensitivity (Yeh et al., 2019) is the only metric that isn't designed to evaluate the *correctness* of an attribution map, but rather the *robustness* of the attribution map against small perturbations. It does this by adding small perturbations to the sample and recomputing the attribution map on the perturbed samples. The maximum value of the L_∞ -norm of the difference between the original and perturbed attribution map is measured. To make different attribution methods comparable, the attribution maps are normalized to unit norm before computing Max-Sensitivity.

$$SENS_{MAX}(\mathbf{x}, m, E, c) = \max_{\|\mathbf{y}-\mathbf{x}\| \leq r} \|E(m, \mathbf{x}, c) - E(m, \mathbf{y}, c)\|$$

Where r is the maximum size of the added perturbation. We choose $r = 0.1$, as in Yeh et al. (2019). Note that this is a type II-metric, meaning that it needs access to the attribution method $E \in \mathcal{E}$ rather than just the attribution map $e \in \mathbb{R}^d$. This metric scales linearly with the number of samples k (here chosen to be 50, as in Yeh et al. (2019)) used to approximate the maximum value, *and* the number of forward/backward passes necessary to compute the attribution map C_{mth} . Note that this can result in very large runtimes when evaluating computationally complex methods.

4.8 Impact coverage

Impact Coverage (Lin et al., 2019) works by applying an adversarial patch to the image, and computing feature attributions on the adversarially attacked image. If the adversarial attack was successful, we would expect a large proportion of the attribution to be inside of the adversarial patch, as the patch caused the model to change its output.

We quantify this by computing the intersection-over-union (IOU) between the k most important pixels according to the attribution map $E(m, \mathbf{x}, c)$ (denoted here as the set T), where k is the number of pixels covered by the patch, and the patch P itself. A score of 1 would indicate that the most important pixels perfectly cover the adversarial patch.

$$COV(\mathbf{x}, m, E, c) = \frac{|T \cap P|}{|T \cup P|}$$

Where P is the set of features that were covered by the adversarial patch. Note that this metric, like Max-Sensitivity, is also a type II metric. Impact Coverage evaluates only the high end of the attribution map. Since the attribution method needs to be executed on the attacked image, this metric has the same complexity as the method being evaluated. Note that an adversarial patch is also needed to compute this metric, meaning that this

complexity is only valid when the adversarial patch is given (that is, when evaluating this metric on a large number of attribution maps for the same model). Impact Coverage can only be computed for image data with a sufficiently high resolution, such that an adversarial patch can be generated successfully.

Impact Coverage stands out from the other metrics because of its causal interpretation. As the adversarial patch was added to the image and the model was re-evaluated, we can be sure that the patch caused the change in the model output. This acts as a form of ground truth, although it is an incomplete form: it is not guaranteed that the entire patch was necessary to change the output. Also, adversarial patches typically have strongly contrasting, high-frequency structure. This means that an attribution method that simply identifies highly contrasting, high-frequency regions in the image will likely score well on this metric, even though it might not be a good explanation of the model's behavior. Therefore, despite the causal interpretation of this metric, a good score is not a completely necessary or sufficient condition for good performance of an attribution method.

4.9 Parameter randomization

The final metric we consider is the Parameter Randomization test (Adebayo et al., 2018). This metric acts as a *sanity check*: rather than scoring each explanation, the results of Parameter Randomization should be interpreted as a pass/fail-test, where passing is a minimal requirement for any method to be considered valuable. The Parameter Randomization test works by randomizing the parameters of the model and recomputing the attribution map for the randomized model. As the attribution map should highlight the features that were important to a specific model, we expect it to be dependent on the model parameters. Therefore, we expect the attribution map to change drastically when the parameters are randomized.

However, Adebayo et al. (2018) warn against a visual inspection of the resulting attribution maps, as it is possible that features with a formerly strongly positive attribution value receive a strongly negative attribution value after randomization. In this case, a visual inspection (which in many cases shows absolute attribution values) can be misleading, as the same features can seem important after randomization, even though their attribution value has changed drastically. Therefore, the change in attribution map is quantified using the absolute value of the Spearman rank correlation coefficient between the attribution maps for the original and the randomized model. If this value is close to zero, then the method is said to pass the sanity check. Note that the authors also introduce variants of this metric using the Structural Similarity Index (Wang et al., 2004) and Histogram of Oriented Gradients (Dalal & Triggs, 2005). However, we will not consider these variants in this work, as they require the image to be divided into patches, which is not always possible (for example, when evaluating attribution methods on tabular data or low-resolution images).

Like Impact Coverage, the Parameter Randomization metric also has a causal interpretation: by randomizing the parameters, we intervene on the model, which allows us to define a form of ground truth. However, recent work suggests some possible limitations of this metric as well. Yona and Greenfeld (2021) model the metric using a causal DAG, and suggest that the task on which the model was trained might act as a confounder in the causal diagram of the metric. This would imply that whether a given explanation method passes or fails the sanity check could depend on the specific task or dataset. Binder et al. (2023) demonstrate that, even after partial randomization of the network, channels with

high activations are still likely to have a strong contribution to the output. For this reason, we only compare the original explanation with an explanation generated for a fully randomized model. Finally, Hedström et al. (2023) show that the similarity metrics employed in Adebayo et al. (2018) are minimized by a statistically uncorrelated random process. This implies that intrinsically noisy explanations, such as gradient-based methods which can be subject to shattered gradient noise (Balduzzi et al., 2017), might be favoured by the Parameter Randomization test.

Note that the underlying assumption from Adebayo et al. (2018), i.e. the idea that any useful attribution method should be sensitive to the model parameters, is not being called into question by any of these works. Instead, the works demonstrate that methods could fail the sanity checks for other reasons than invariance to model parameters, and the outcome of the sanity check might depend on the specific dataset. For this reason, we will also compute the sanity check for each dataset separately.

5 Attribution methods

An overview of the attribution methods included in this study can be seen in Table 2. We divide the methods in three types: Gradient-based, CAM-based and Perturbation-based. We also mention if the method requires the model to be differentiable, convolutional, or if it has no requirements about the model. The implementation used for GradCAM++ (Chattopadhyay et al., 2018) and ScoreCAM (Wang et al., 2020) is available in the torch-cam package (Fernandez, 2020). For XRAI (Kapishnikov et al., 2019), an implementation is available in the Saliency package provided by PAIR.¹ For all other methods, the Captum package (Kokhlikyan et al., 2020) was used. For more details on the methods, we refer to the original papers in Table 2.

6 Experimental setup

In this section, we describe the datasets used in the experiments, the different implementations of feature masking, and the methods of statistical analysis that we performed on the metric scores. For a demonstration of the methodology on tabular datasets, see Appendix C.

6.1 Datasets

All experiments were conducted on 14 attribution methods and 8 datasets. The datasets can be divided into three groups:

- Low-dimensional datasets (28x28x1): MNIST (Lecun et al., 1998), FashionMNIST (Xiao et al., 2017)
- Medium-dimensional datasets (32x32x3): CIFAR-10, CIFAR-100 (Krizhevsky, 2009), SVHN (Netzer et al., 2011)
- High-dimensional datasets (224x224x3): ImageNet (Deng et al., 2009), Caltech-256 (Griffin et al., 2022), Places-365 (Zhou et al., 2017)

¹ <https://github.com/pair-code/saliency>.

Table 2 Summary of methods

Method	Type	Complexity	Model requirements
Gradient (Simonyan et al., 2014)	Gradient	$\mathcal{O}(1)$	Differentiable
InputXGradient (Shrikumar et al., 2017)	Gradient	$\mathcal{O}(1)$	Differentiable
Deconvolution (Zeiler & Fergus, 2014)	Gradient	$\mathcal{O}(1)$	Differentiable
Guided Backpropagation (Springenberg et al., 2014)	Gradient	$\mathcal{O}(1)$	Differentiable
DeepLIFT (Shrikumar et al., 2017)	Gradient	$\mathcal{O}(1)$	Differentiable
Integrated Gradients (Sundararajan et al., 2017)	Gradient	$\mathcal{O}(n)$	Differentiable
XRAI (Kapishnikov et al., 2019)	Gradient	$\mathcal{O}(n)$	Differentiable
Expected Gradients (Erion et al., 2021)	Gradient	$\mathcal{O}(nm)$	Differentiable
SmoothGrad (Smilkov et al., 2017)	Gradient	$\mathcal{O}(m)$	Differentiable
VarGrad (Adebayo et al., 2018)	Gradient	$\mathcal{O}(m)$	Differentiable
DeepSHAP (Lundberg & Lee, 2017)	Gradient	$\mathcal{O}(m)$	Differentiable
KernelSHAP (Lundberg & Lee, 2017)	Perturbation	$\mathcal{O}(m)$	None
LIME (Ribeiro et al., 2016)	Perturbation	$\mathcal{O}(m)$	None
GradCAM (Selvaraju et al., 2017)	CAM	$\mathcal{O}(1)$	Convolutional
Guided GradCAM (Selvaraju et al., 2017)	CAM	$\mathcal{O}(1)$	Convolutional
GradCAM++ (Chattopadhyay et al., 2018)	CAM	$\mathcal{O}(1)$	Convolutional
ScoreCAM (Wang et al., 2020)	CAM	$\mathcal{O}(c)$	Convolutional

Complexity is expressed as number of executions of the model. n , m and c are path length, number of perturbed samples/baselines, and number of channels in the final layer, respectively. n and m are hyperparameters of the method, c depends on the model being explained. Note that, even though many methods have the same asymptotic complexity, the typical values of hyperparameters can vary a lot, for example DeepSHAP usually needs much fewer samples than KernelSHAP or LIME, making it computationally less expensive.

For the low-dimensional datasets, a simple CNN architecture (2 convolutional layers with 32 and 64 channels, followed by a fully connected hidden layer with 128 nodes) was trained. For the medium- and high-dimensional datasets, we used Resnet20 and Resnet18,

respectively. The models for the low- and medium-dimensional datasets were trained up to a test set accuracy of at least 90%, except for CIFAR-100, where a top-five accuracy of 90.6% was reached. For Caltech-256 and Places-365, the models were trained up to a top-five test set accuracy of 91.6% and 83.7%, respectively. For ImageNet, the built-in Resnet18 model of torchvision² was used, obtaining a top-five accuracy of 89.08%. The metric scores were computed for all attribution methods on 256 correctly-classified samples for each dataset. Note that an adversarial patch was only generated for the high-dimensional datasets (ImageNet, Caltech-256, Places-365), which means that the Impact Coverage could only be computed for these datasets.

6.2 Masking

Except for Infidelity, Impact Coverage, and the Parameter Randomization test, all metrics depend in some way on the *masking* of features to remove information. When masking features, we try to replace the feature value with some “neutral” value that is expected to remove the original information contained in the feature. However, the choice of this neutral value is not obvious (Sturmels et al., 2020). We consider three options:

- *Dataset mean* The first and simplest way of masking is by replacing the feature with a constant zero value (in the case of color images, we do the same for each color channel). Since the data is z -normalized to have $\mu = 0$ and $\sigma = 1$, this is equivalent to changing the feature into the average feature value over the training dataset. A disadvantage of this technique is that, if the original feature was already close to the average value, the value remains nearly unchanged after masking and the information might not be properly destroyed. Specifically for image data, masking out large regions with a constant value can preserve some of the spatial information in the image. Additionally, masking features with any constant value can introduce high-frequency artifacts to the image in question, driving the input away from the data manifold (Fong & Vedaldi, 2017).
- *Uniform random* To mitigate some of the problems of the dataset mean value, we can also draw values from a standard uniform distribution $\mathcal{U}(0, 1)$. That way, masked out features are less likely to remain unchanged after masking, and spatial information is likely to be successfully destroyed if larger regions of the image need to be masked. However, using a uniform distribution to mask out features introduces even more adversarial high-frequency artifacts than using a constant dataset mean.
- *Blur* To reduce the high-frequency artifact problem of the first two masking procedures, pixels can instead be masked out using blurring (Fong & Vedaldi, 2017). We use the OpenCV normalized box filter (Bradski, 2000) with kernel size $k = 0.5$ to produce a blurred version of the original image. Pixels are then masked out by replacing them by their blurred equivalents. Although this technique mitigates the high-frequency artifact problem, it again has the disadvantage that spatial information might not be completely destroyed after masking.

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

6.3 Statistical analysis

In this section we provide a brief overview of the statistical techniques used to analyse our results. We first use a paired t-test to identify which methods outperform a basic random baseline on the metrics. Next, we compute the correlations of scores between different metrics, which allows us to measure which metrics are or are not measuring the same underlying properties. We then study the consistency of method rankings as given by each metric, which can be viewed as a quality check for the metrics themselves. Finally, we propose a technique to compare two methods in more detail.

6.3.1 Statistical significance testing

For each metric and each method, we use a paired t-test to verify if the method performs significantly better than a uniform baseline on the given metric. More specifically, we test if the difference in metric score for the explanation method is significantly larger/smaller (depending on the metric in question) than the score obtained by the uniform baseline. The uniform baseline is defined as a “pseudo-method”, which simply assigns random values $u \sim \mathcal{U}(0, 1)$ to each feature. This baseline is computed once for every input sample, such that the same baseline attribution map is compared to each of the attribution maps computed by the explanation methods. Note that a different, more informative baseline method could also be used. For example, a simple edge detection algorithm could be used to establish a more competitive baseline, while retaining the property that any valid explanation method should be expected to outperform it. Alternatively, an existing explanation method could be used as the baseline, for example to test whether some other method specifically outperforms that baseline method. We leave a further investigation of different baseline methods to future work.

Because for each metric, multiple methods are tested against the random baseline, we use Bonferroni multiple testing correction (Bland & Altman, 1995). If the result of the test is significant after correction ($p < 0.01$), we report the Cohen’s d effect size for paired t-tests (Cohen, 1988):

$$\frac{\mu_d}{\sigma_d}$$

which is simply the average difference in metric scores divided by the standard deviation of the differences. Since the absolute values of most metrics carry little to no semantic meaning, these effect sizes are only relevant relative to each other. For this reason, the effect sizes are scaled to $[0, 1]$ for each metric, such that the best-performing method has an effect size of 1.

6.3.2 Inter-metric correlation

Inter-metric correlations are computed as the Spearman rank correlation between metric scores, averaged over all methods (except the random baseline). These correlations allow us to identify which metrics are measuring different underlying aspects, and which metrics are mutually redundant.

6.3.3 Ranking consistency

Ranking consistency assesses how consistent a metric is in ranking the methods across the different images. This is measured using Krippendorff's α (Krippendorff, 2019). Krippendorff's α is a statistic usually used to measure *inter-rater reliability*: the degree to which different *raters* (for example, for a psychological test) agree in their assessments. Krippendorff's α is defined as follows:

$$\alpha = 1 - \frac{D_o}{D_e}$$

where D_o is the *observed disagreement*, and D_e is the *disagreement expected by chance*. If $\alpha = 1$, then the ranking is perfectly consistent: the ranking of methods produced by the metric is identical for each image. If $\alpha = 0$, then the ranking is completely random. For more details on how these values are computed, we refer the reader to Krippendorff (2011).

6.3.4 Pairwise comparison using PoS

Once a global overview of method performance has been established using the statistical significance test (Sect. 6.3.1), two or more methods can be selected for a more detailed comparison. Such a comparison is then made by performing a new statistical test, this time comparing the methods to each other, rather than to a trivial random baseline.

In this case, we use the Power of Superiority (PoS) effect size to measure the difference between methods. This measure is simply the fraction of images where method A outperforms method B. This effect size measure is less informative when comparing methods to the random baseline, as we expect methods to at least consistently outperform the baseline, leading to a saturated effect size of 1. If two methods are selected that are more similar in their performance, the PoS can give an intuition to how often one method (usually a more computationally complex one) outperforms the other.

For example, if the difference in metric scores is very large, but the PoS is only slightly larger than 0.5, this would mean that method A outperforms method B only in a small majority of images. If computational cost is a concern, this can make it more interesting to choose for the computationally cheaper method.

7 Results

In this section, we describe the results of paired t-tests, inter-metric correlations, and ranking consistency of metric scores on the different datasets. Masking is done using the dataset mean approach unless stated otherwise. Finally, we perform a pairwise comparison of DeepSHAP and DeepLIFT on MNIST, CIFAR-10 and ImageNet.

7.1 Paired t-tests

The results of the paired t-tests are shown in Figs. 1, 2 and 3, for the low-, medium- and high-dimensional datasets, respectively. For each method-metric pair, a square is drawn if the result of the paired t-test is significant after Bonferroni correction for multiple testing

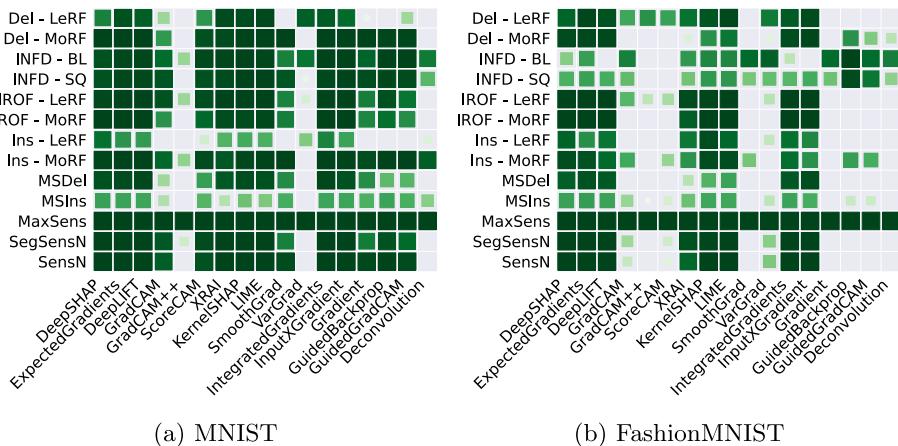


Fig. 1 Results of paired t-tests (low-dimensional datasets). A square is only drawn if the corresponding result was significant after Bonferroni correction ($p < 0.01$)

($p < 0.01$), with the size and color of the square indicating the effect size (Cohen's d). Effect sizes are normalized such that a value of 1 corresponds to the largest effect size for a given metric.

On MNIST, most methods significantly outperform the random baseline on nearly all metrics. On FashionMNIST however, CAM-based methods, Guided Backpropagation, Deconvolution, SmoothGrad and VarGrad perform significantly worse than the others.

On the medium-dimensional datasets, we see more complementarity in the results, although this still depends on the dataset. On CIFAR-10 and CIFAR-100, we notice that the CAM-based methods along with XRAI, KernelSHAP and LIME perform very similarly, with this group of metrics outperforming DeepSHAP, DeepLIFT and ExpectedGradients on some metrics and vice versa. This similarity in behaviour can be linked to the fact that these methods produce more coarse-grained attribution maps, as all CAM-based methods rely on upsampling the final convolutional layer, and XRAI, KernelSHAP and LIME rely on image segmentation. Conversely, DeepLIFT, DeepSHAP and ExpectedGradients are all based on modified versions of the gradient, which tends to produce very granular attribution maps. This difference in granularity could be the source of the observed complementarity. For SVHN, a number of methods significantly outperform the baseline across all metrics.

In the high-dimensional case, we see fewer differences between the datasets. The same complementarity between the coarse-grained and fine-grained methods is again noticeable for all three datasets, suggesting that it is linked to the complexity or dimensionality of the classification problem.

We also note that the coarse-grained methods tend to outperform the others on Impact Coverage (COV), which could only be computed for high-dimensional datasets. Interestingly, the results for Impact Coverage seem to be complementary to those of Deletion-MoRF and Minimal Subset Deletion. A possible explanation is the fact that Impact Coverage makes the implicit assumption that the entire adversarial patch is equally important, which is not necessarily the case, as discussed in Sect. 4.8. This might bias the metric towards coarse-grained attribution methods. If only a few pixels in the adversarial patch are truly important, then fine-grained attribution maps might highlight only those few pixels,

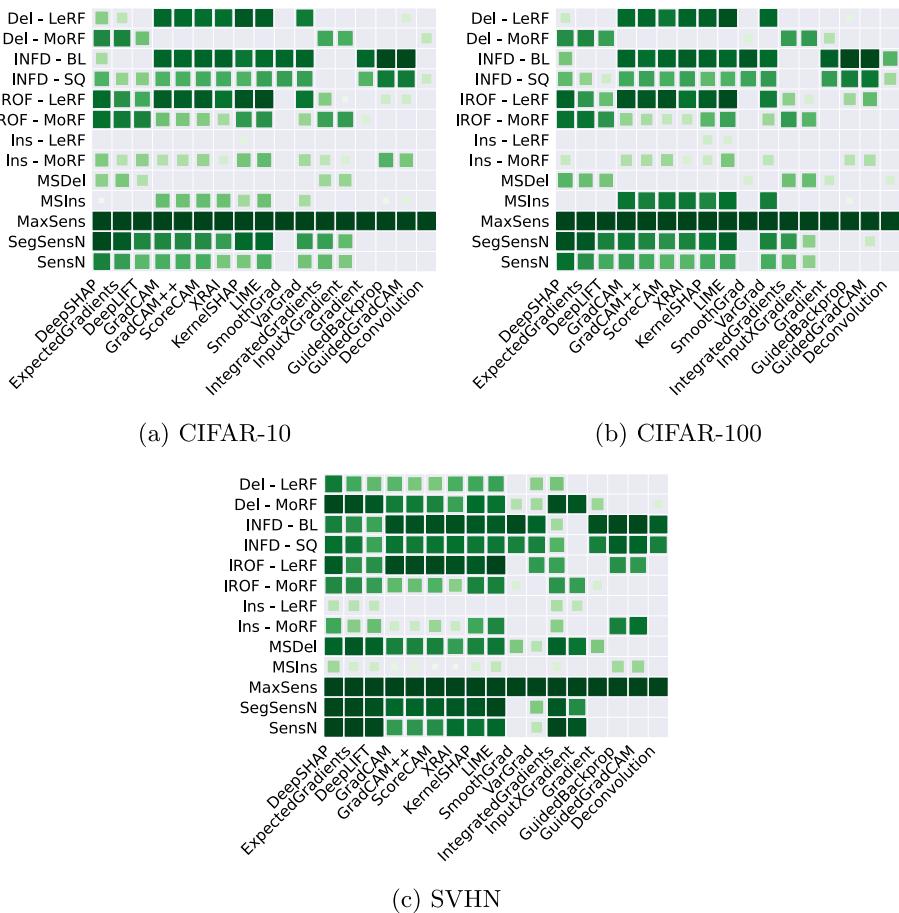


Fig. 2 Results of paired t-tests (medium-dimensional datasets). A square is only drawn if the corresponding result was significant after Bonferroni correction ($p < 0.01$)

resulting in a low Impact Coverage score. Further research is needed to confirm or refute this hypothesis.

We draw three conclusions from these results:

- (1) Depending on the dataset, very simple and computationally cheap methods can perform nearly as well as computationally more expensive methods.
- (2) Complementarity between methods, where some methods outperform other methods on a subset of metrics and vice versa, suggests that a combination of attribution maps given by different methods might provide more information than the individual attribution maps. This is related to the idea proposed in Tomsett et al. (2020) that different metrics might be measuring different underlying aspects of the attribution maps.
- (3) The complementarity of results between coarse-grained and fine-grained attribution methods suggests that these methods might have to be evaluated in fundamentally different ways, focusing on single-pixel importance for fine-grained maps and on a more high-level view for coarse-grained maps.

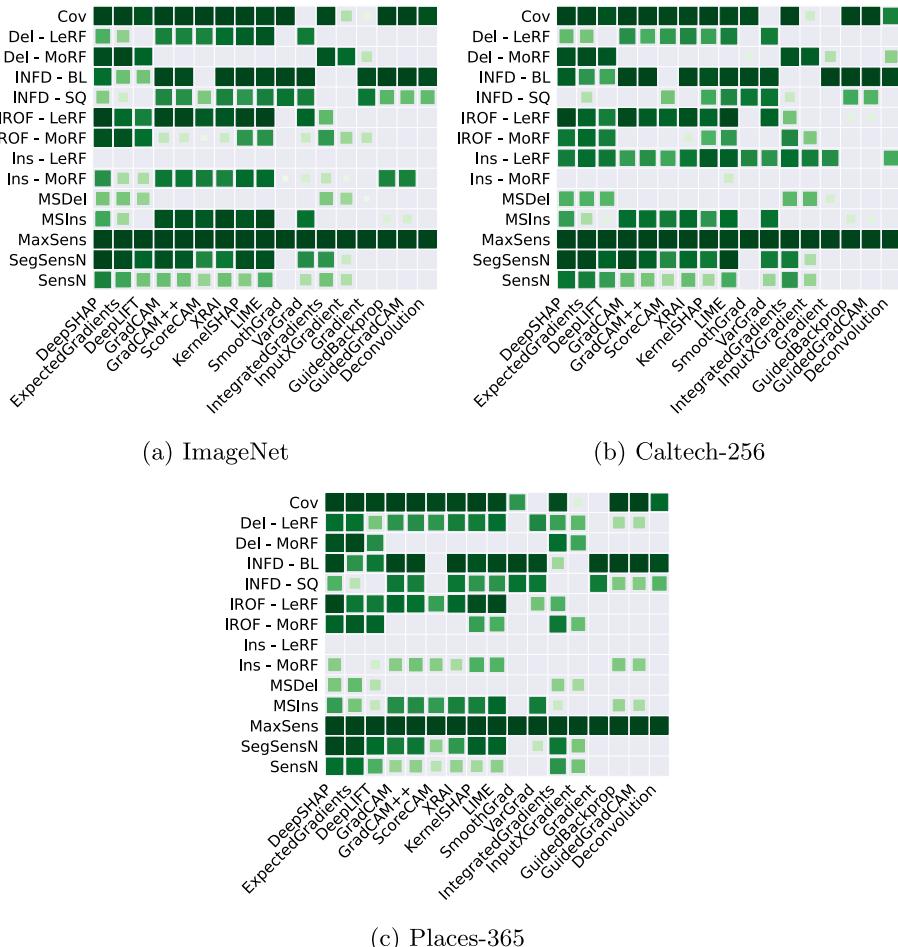


Fig. 3 Results of paired t-tests (high-dimensional datasets). A square is only drawn if the corresponding result was significant after Bonferroni correction ($p < 0.01$)

7.2 Inter-metric correlations

Figure 4 shows average inter-metric correlations for the low-, medium- and high-dimensional datasets. For specific correlations per dataset, see Figure 14. In general, we note similar patterns of correlations for the three dimensionalities. Most metrics have relatively low correlations, suggesting that they might be measuring different underlying aspects of the attribution maps, as proposed in Tomsett et al. (2020). We also note strong negative correlations between certain pairs of metrics, more specifically MoRF/LeRF-pairs, which suggests that MoRF/LeRF-pairs contain largely redundant information. This insight can be used to reduce computational cost in future benchmarking efforts, by selecting only MoRF or LeRF metrics. Interestingly, Fig. 4c shows very little correlation between the Impact Coverage and Parameter Randomization metrics.

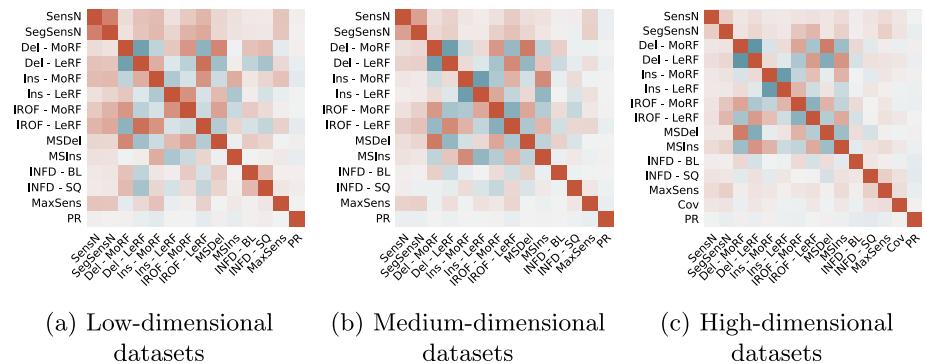


Fig. 4 Average inter-metric correlations for low-, medium- and high-dimensional datasets. Impact Coverage (Cov) was only computed for the high-dimensional datasets due to the requirement of an adversarial patch (see Sect. 4.8)

This is notable, as both of these metrics have a causal interpretation, even though those causal interpretations are different: Impact Coverage intervenes on the data, whereas Parameter Randomization intervenes on the model. Finally, we note that correlations between segmented and non-segmented metrics (for example, Deletion and IROF) are stronger for low-dimensional datasets. This is to be expected, since the low dimensionality of the data causes segments to be composed only of a few pixels.

Table 3 shows inter-metric correlations of different metric implementations on ImageNet (results on the other datasets were generally similar). We note that, although different metrics have relatively low correlations, correlations between different implementations of the same metric are generally quite high. We can conclude from this that different implementations of the same metric generally provide redundant information. We recommend first deciding which masking procedure makes most sense for a given dataset and/or model, rather than performing full measurements using a large number of masking procedures.

Table 3 Inter-metric correlations of different implementations of metrics on ImageNet. C, B and R stand for Constant, Blur and Random masking, respectively

<i>Del_{MoRF}</i>			<i>Del_{LeRF}</i>			<i>Ins_{MoRF}</i>			<i>Ins_{LeRF}</i>		
C	B	R	C	B	R	C	B	R	C	B	R
C	1.00			1.00			1.00			1.00	
B	0.87	1.00		0.83	1.00		0.89	1.00		0.89	1.00
R	0.85	0.75	1.00	0.82	0.69	1.00	0.54	0.60	1.00	0.56	0.62
<i>IROF_{MoRF}</i>			<i>IROF_{LeRF}</i>			<i>MS_{Del}</i>			<i>MS_{Ins}</i>		
C	B	R	C	B	R	C	B	R	C	B	R
C	1.00			1.00			1.00			1.00	
B	0.92	1.00		0.90	1.00		0.81	1.00		0.78	1.00
R	0.79	0.76	1.00	0.75	0.72	1.00	0.83	0.70	1.00	0.70	0.60

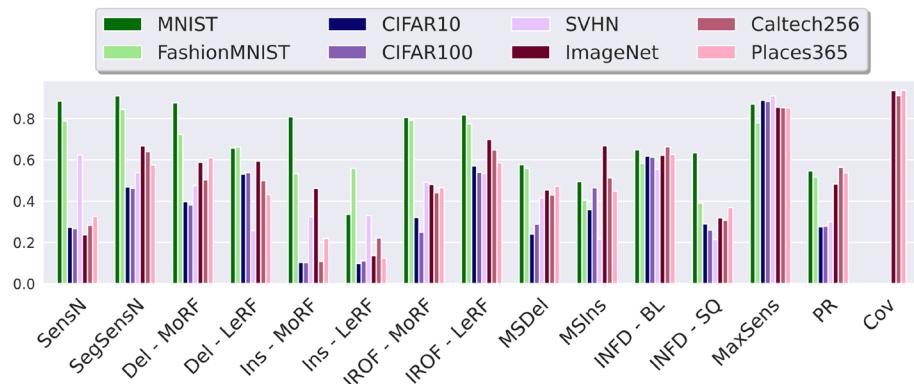


Fig. 5 Krippendorff's α for default implementations of different metrics on all datasets. Low-, medium- and high-dimensional datasets are indicated in green, blue and red tones, respectively. Impact Coverage (Cov) was only computed for the high-dimensional datasets due to the requirement of an adversarial patch (see Sect. 4.8) (Color figure online)

7.3 Ranking consistency

The values of α for all datasets are shown in Fig. 5. It can be observed that most of the metrics are most consistent on the low-dimensional datasets (MNIST, FashionMNIST). Impact Coverage was only measured for high-dimensional datasets because of the reliance on an adversarial patch, and has the highest values of α . We also see that there is no clear pattern between the medium- and high-dimensional datasets, implying that α doesn't simply decrease with increasing dimensionality. We note that our proposed segmented variant of Sensitivity-n has a higher α for high-dimensional datasets, confirming the intuition that this metric has a higher signal-to-noise ratio for high-dimensional data.

Although there are some metrics that have a significantly lower α value across all or almost all datasets, such as Ins_{MoRF} , Ins_{LeRF} , there is no clear subset of metrics that is generally superior to all others in terms of ranking consistency, with two exceptions: Max-Sensitivity and Impact Coverage. However, Max-Sensitivity measures robustness of explanations rather than correctness, and Impact Coverage can only be computed for high-dimensional datasets. From these results, we conclude that the ideal subset of metrics to measure depends on the dataset and model. Different implementations of the same metric (using different masking procedures) generally have similar values for α . An overview of Krippendorff α for all metric implementations is given in Appendix A.

7.4 Pairwise comparison of methods

We use the proposed framework in Sect. 6.3.4 to compare the performance of DeepSHAP and DeepLIFT on MNIST, CIFAR-10 and ImageNet. We choose these two methods because they have very similar results across all datasets in Figs. 1, 2 and 3, which is to be expected as DeepSHAP is based on DeepLIFT. However, DeepSHAP is computationally much more expensive than DeepLIFT, so if the fraction of images where it outperforms DeepLIFT is relatively small, it might not be worth the cost. Note that this specific choice

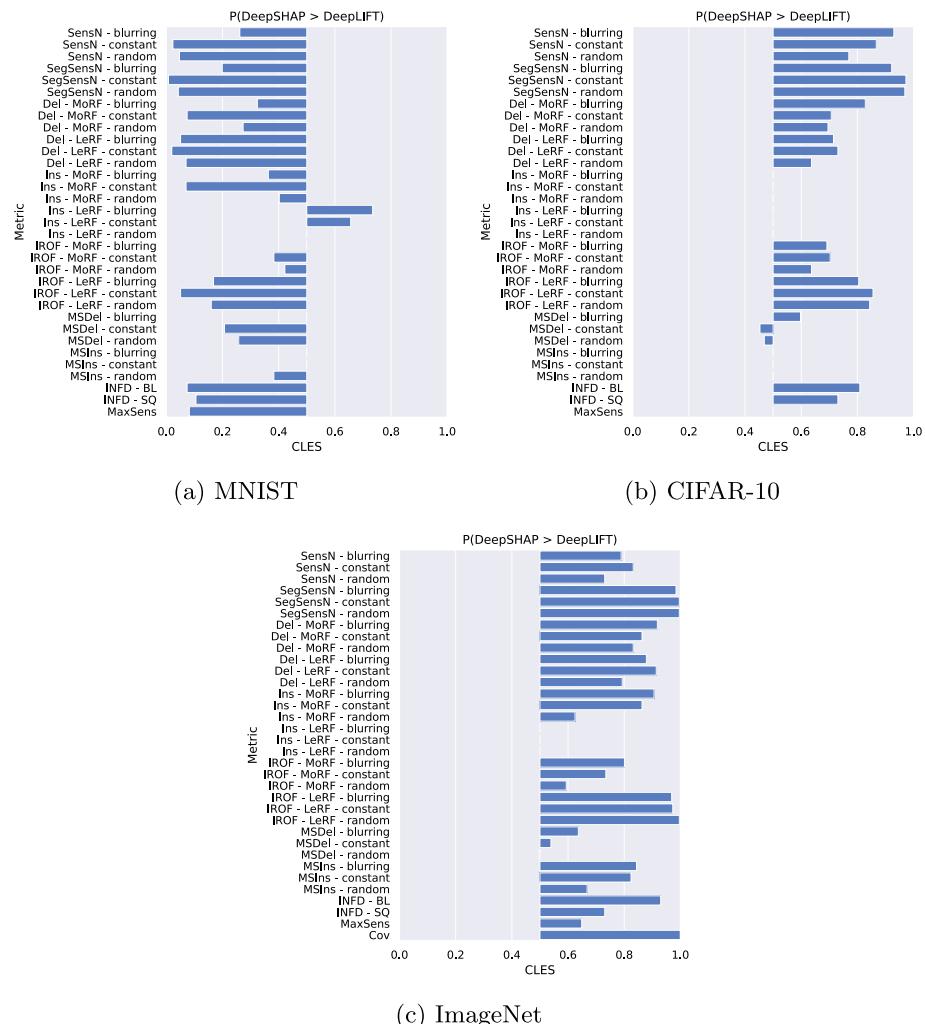


Fig. 6 Comparison of DeepSHAP versus DeepLIFT using Common Language Effect Size

was made merely for demonstration purposes. In practice, we recommend that practitioners select methods to compare based on their results on the paired t-tests and other relevant factors such as computational complexity, difficulty of implementation, etc.

The results are shown in Fig. 6. Each bar corresponds to a paired t-test between the results for DeepSHAP and DeepLIFT on a single metric. A bar is only drawn if the corresponding result was significant ($p < 0.01$). The bars are centered on 0.5, since a Probability of Superiority of 0.5 would indicate that both methods are equivalent, each outperforming the other in 50% of cases.

We see that, although performance in terms of absolute metric scores is very similar between the two methods (as shown in Sect. 7.1), the Probability of Superiority (PoS)

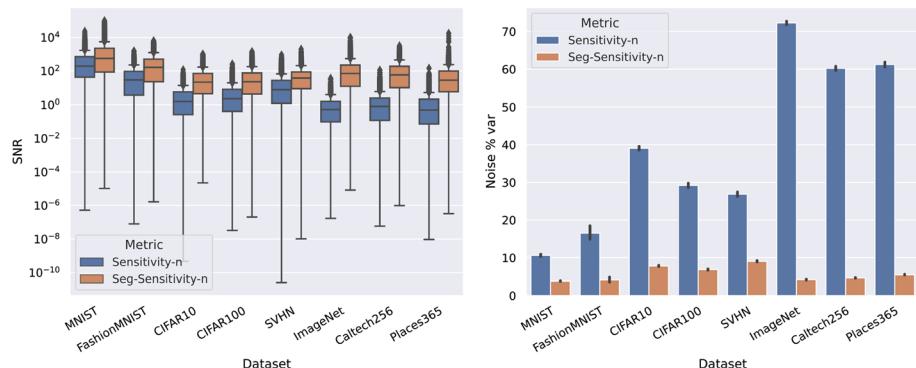


Fig. 7 Comparison between Sensitivity-*n* and Seg-Sensitivity-*n*. Left: Signal-to-noise ratio (logarithmic scale). Right: noise fraction of variance

varies greatly depending on the dataset. On ImageNet, DeepSHAP outperforms DeepLIFT for most images, with the PoS ranging between 60–80% for most metrics. On CIFAR-10 however, the difference between the two methods is much smaller. Finally, on MNIST, DeepSHAP is outperformed by DeepLIFT on a majority of images, for almost all metrics. This indicates that the relative performance of methods is strongly dependent of the dataset in question.

7.5 Sensitivity-*n* versus seg-sensitivity-*n*

To compare our proposed metric Seg-Sensitivity-*n* to the original Sensitivity-*n*, we measure the stability of both metrics in two ways. First, we measure the signal-to-noise ratio (SNR) of both metrics. We repeatedly compute both Seg-Sensitivity-*n* and Sensitivity-*n* scores 100 times on 256 images (where the same images were used for both metrics). We then compute the SNR ratio of the metric for each image as $\frac{\mu^2}{\sigma^2}$, where μ is the mean of the 100 metric values, and σ is the standard deviation. The results are shown in the left part of Fig. 7. Note that the SNR of Seg-Sensitivity-*n* for high-dimensional datasets (ImageNet, Caltech-256 and Places-365) is significantly higher than the SNR of Sensitivity-*n*. On the other datasets, the SNR is also larger for Seg-Sensitivity-*n*, although the difference is smaller.

A different way to measure the stability is to look at the *noise fraction of variance*. To compute this, we compute the ratio of the within-sample variance (the variance of the 100 repeated measurements for each sample) to the between-sample variance (the total variance of all measurements on all samples). A low noise fraction of variance corresponds to a clear signal. These results are shown in the right plot of Fig. 7. We see again that the noise fraction of variance for Sensitivity-*n* is much larger than for Seg-Sensitivity-*n*, especially on the high-dimensional datasets.

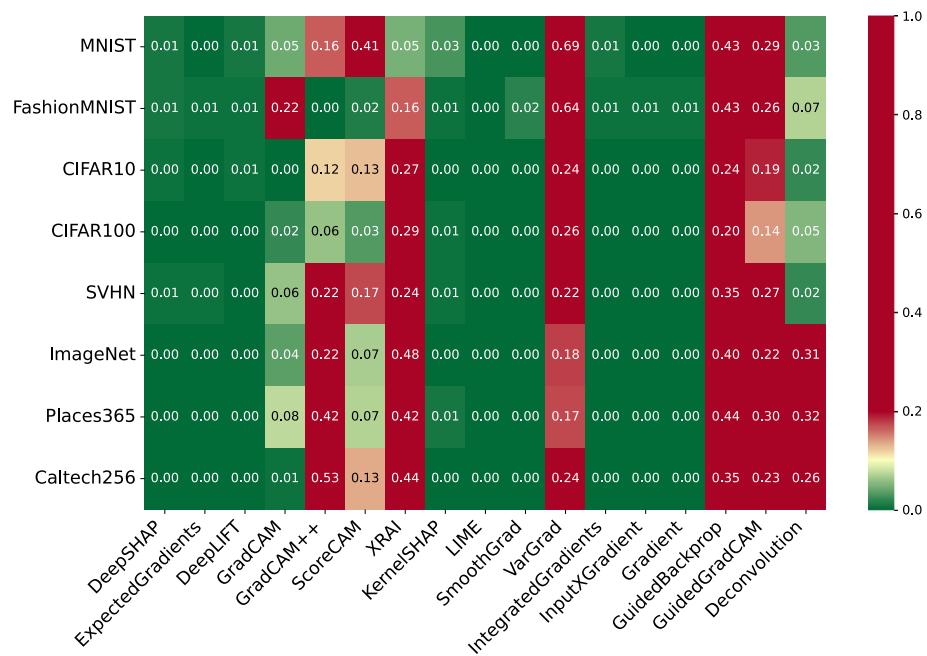


Fig. 8 Results of the Parameter Randomization metric

7.6 Parameter randomization

Results of the Parameter Randomization test are shown in Fig. 8. We consider any method that has an absolute rank correlation larger than 0.2 as failing the test. Note that these rank correlations are between the attribution maps produced by the same method before and after randomization of the model parameters. They should not be confused with the inter-metric correlations discussed in Sect. 7.2.

We notice that the methods that were identified in (Adebayo et al., 2018) as failing the test (Guided Backpropagation and Guided Grad-CAM) also have relatively large correlation values across all datasets, implying that they indeed fail the sanity checks on all these datasets. Additionally, methods that were shown by Adebayo et al. (2018) to pass the test (Gradient, Integrated Gradients and InputXGradient) obtain low correlation scores across all datasets as well. We can therefore confirm many of the experimental findings of Adebayo et al. (2018) across a wider selection of datasets.

Interestingly, ScoreCAM and XRAI have large correlation values across many datasets, even though they were described as passing the Parameter Randomization test by their original authors (Wang et al., 2020; Kapishnikov et al., 2019). However, ScoreCAM was only inspected visually, and the test for XRAI was only conducted on MNIST. Our findings confirm that XRAI passes the test on MNIST, but show that the same method also fails on other datasets. We therefore conclude that the outcome of the Parameter Randomization test is indeed dependent on the dataset and/or model, which is in concordance with the findings of Yona and

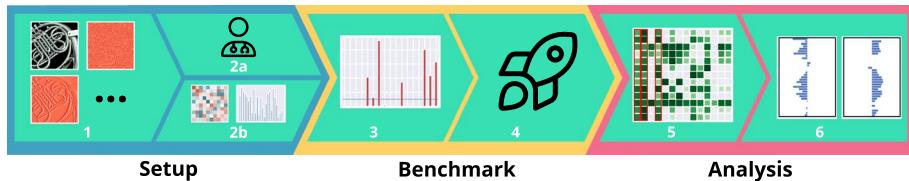


Fig. 9 Visual overview of the proposed benchmarking guidelines. (1) Choose a baseline. This can be a random baseline, an edge detector, or some other baseline of choice. (2) Select metrics. This can be done manually (2a) or through a pilot study (2b) where metrics are selected based on inter-metric correlations and Krippendorff α . (3) Compute the parameter randomization sanity check and discard methods that fail it. (4) Run the remaining methods on the selected metrics. (5) Perform statistical tests against the random baseline and make a selection of well-performing methods. (6) Perform pairwise comparisons between the selected methods

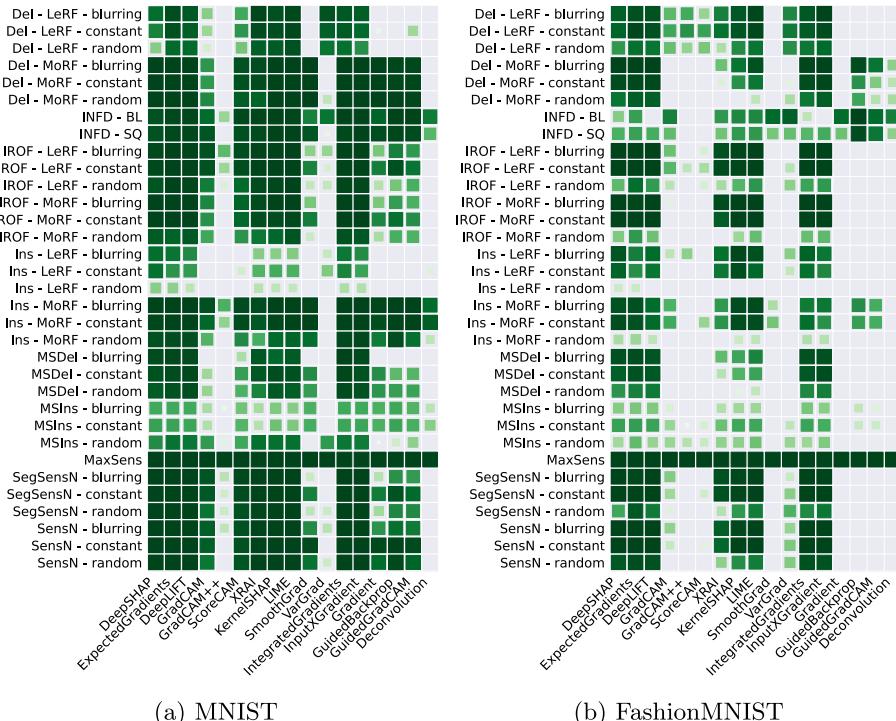


Fig. 10 Results of paired t-tests for all metrics on low-dimensional datasets

Greenfeld (2021). Based on these results, our recommendation to practitioners is to perform a quantitative analysis of the Parameter Randomization test on the specific dataset in question, rather than performing a simple visual inspection or assuming that a method will pass the test if it was shown to pass the test on a different dataset (such as MNIST).

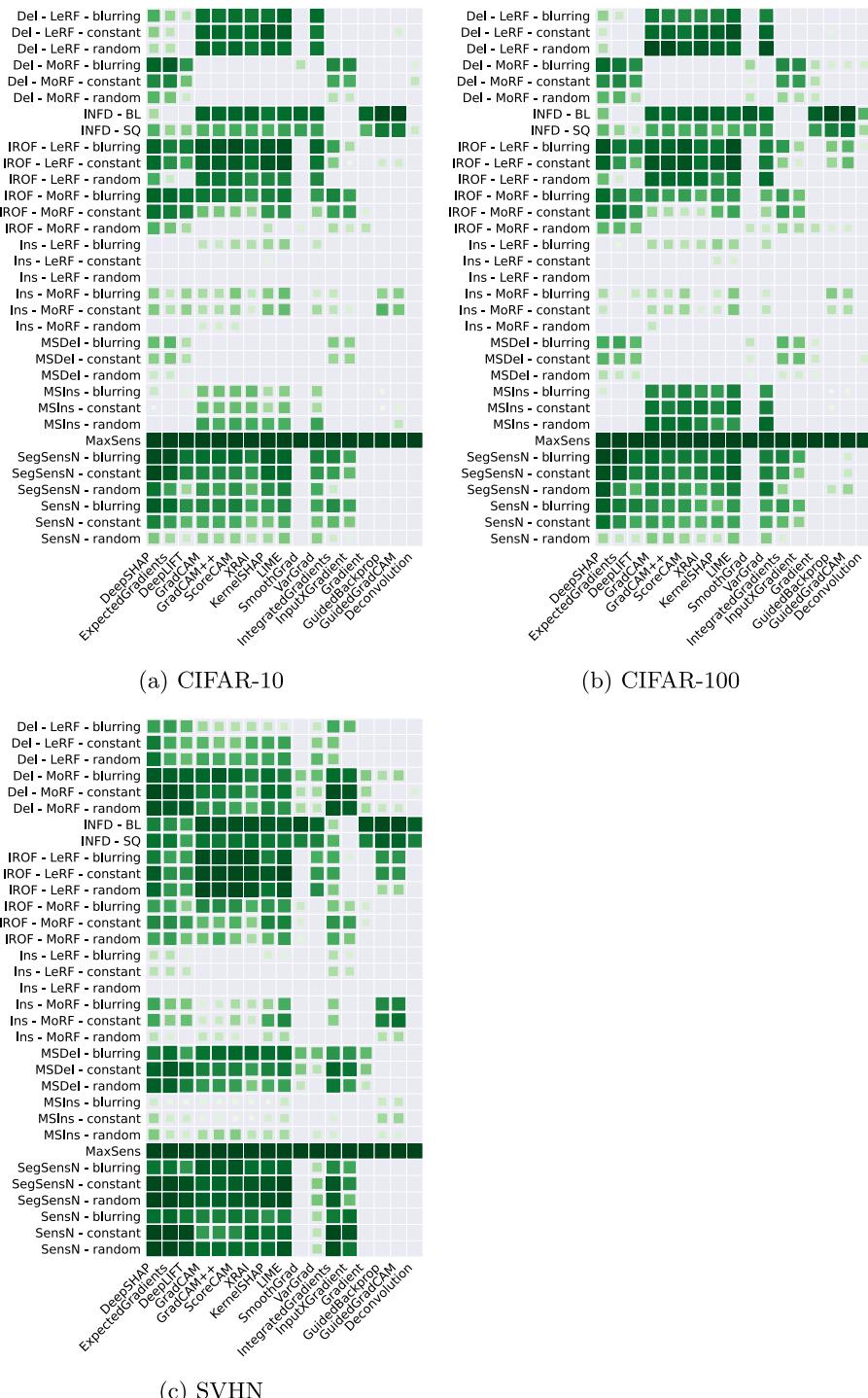


Fig. 11 Results of paired t-tests for all metrics on medium-dimensional datasets

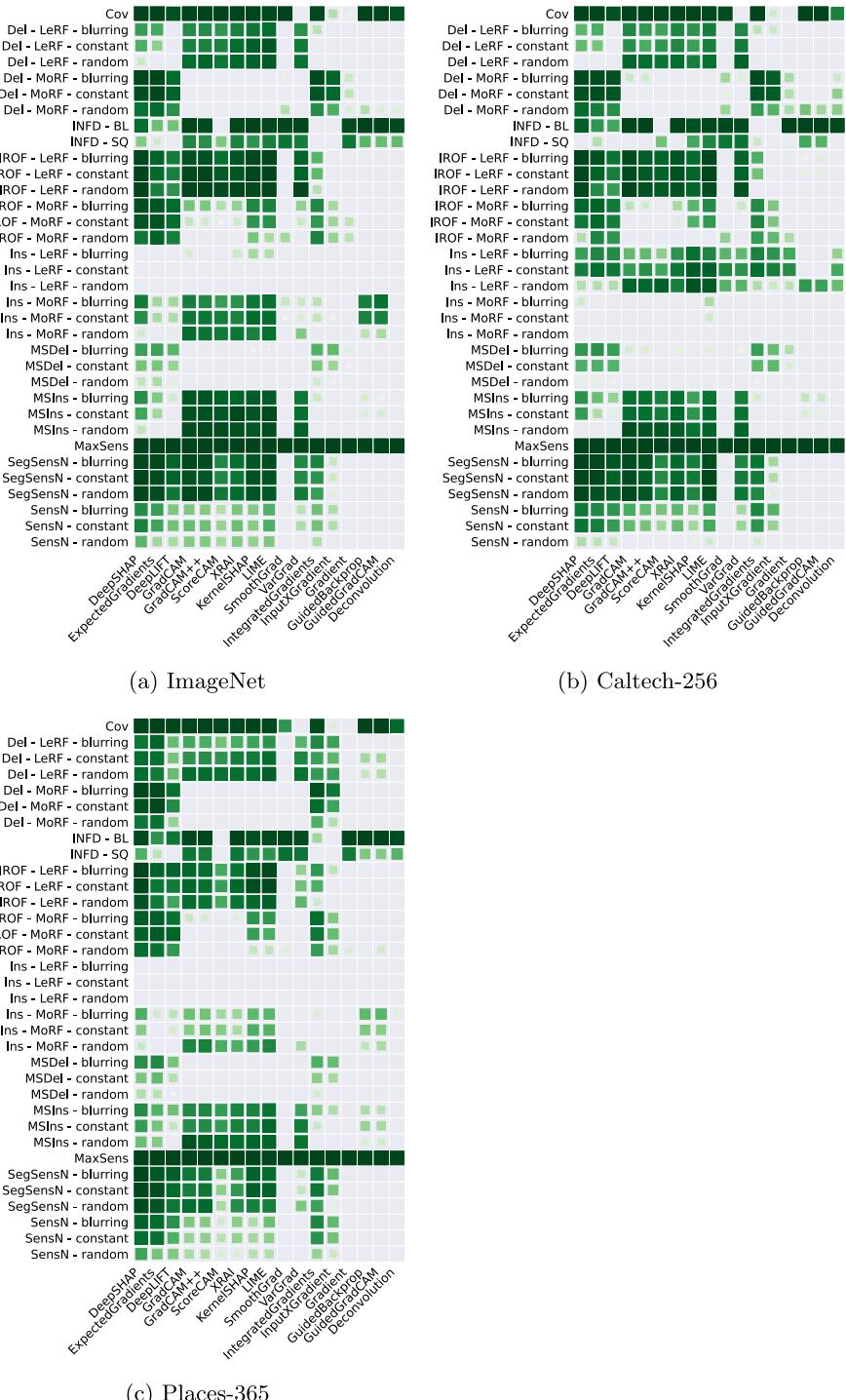


Fig. 12 Results of paired t-tests for all metrics on high-dimensional datasets

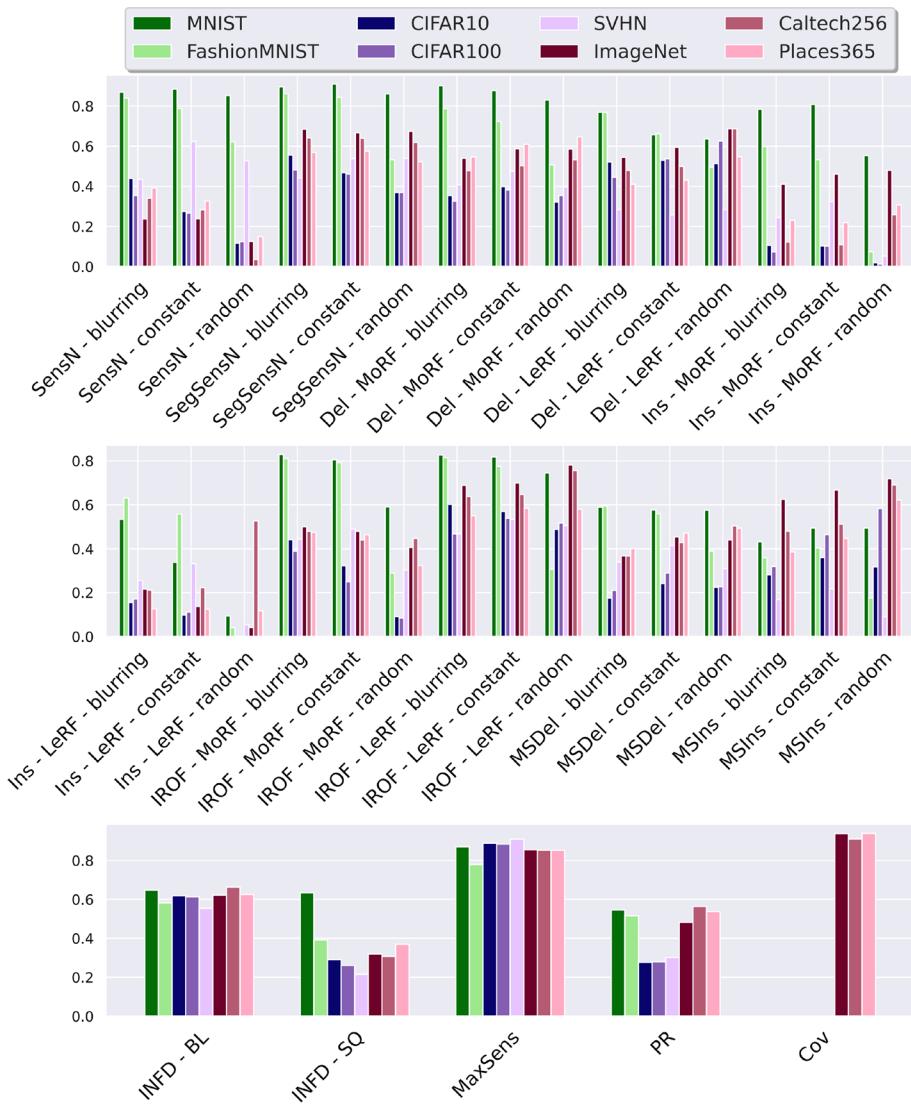


Fig. 13 Krippendorff's α for all implementations of all metrics on all datasets. Low-, medium- and high-dimensional datasets are indicated in green, blue and red tones, respectively (Color figure online)

8 Conclusion and future work

We have performed an extensive study of the behaviour of a large number of attribution metrics and methods, on a collection of image datasets with varying complexity and dimensionality. From this investigation, we draw the following general conclusions:

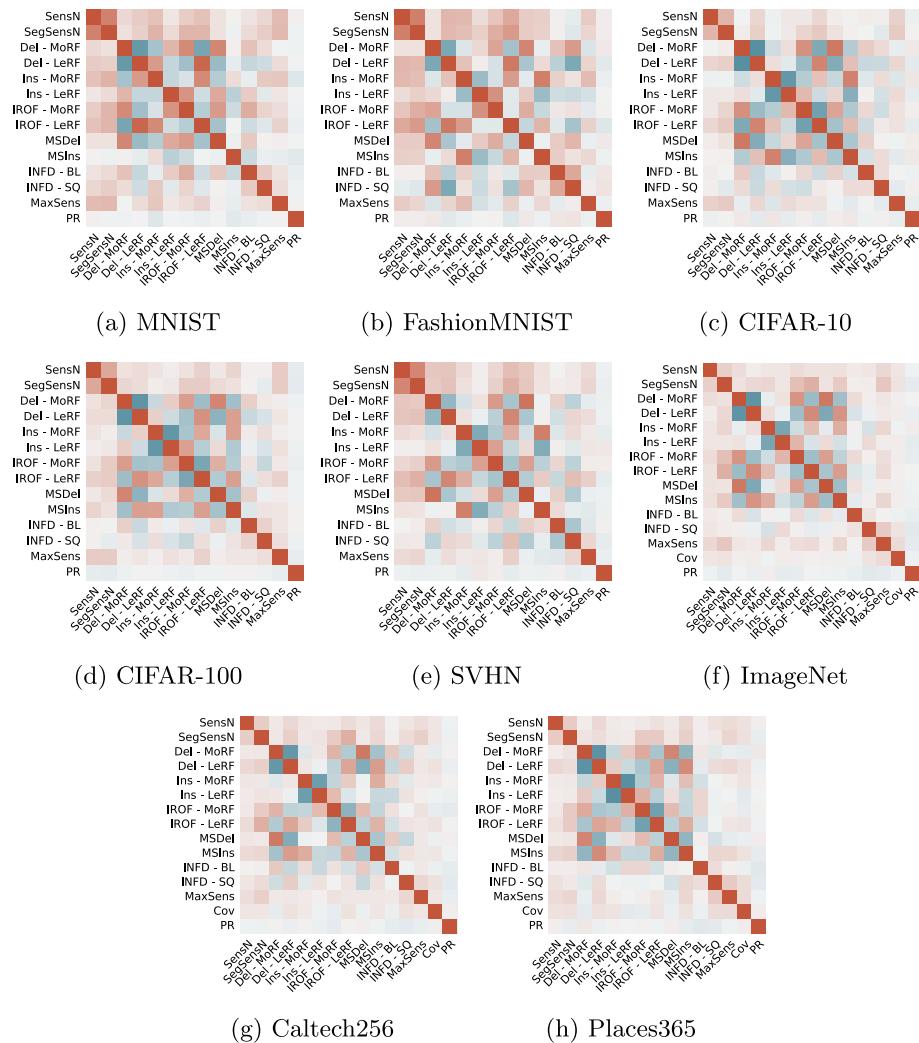


Fig. 14 Inter-metric correlations for all datasets

- Metric scores vary strongly for different datasets. This implies that the performance of attribution methods should be measured for each specific use case, rather than drawing general conclusions from the results on a set of benchmark datasets.
- Most metrics tend to have low ranking consistency, shown by the relatively low values of Krippendorff α . From this we conclude that a rigorous statistical testing approach is necessary to draw any dataset-wide conclusions. Also, the ranking consistency values of metrics themselves are not consistent across datasets, implying that there is no generally superior evaluation metric in terms of ranking consistency.
- We confirm the conclusion from Tomsett et al. (2020) that metrics do not necessarily measure the same underlying concept to a larger amount of metrics, and extend their findings to include Sensitivity-n (Ancona et al., 2018), Infidelity (Yeh et al.,

2019), IROF (Rieger & Hansen, 2020) and Impact Coverage (Lin et al., 2019). This can be seen in the low inter-metric correlation values between these metrics.

- The result of the Parameter Randomization test, introduced as a sanity check in AdEBAYO et al. (2018), is dataset-dependent, meaning that whether a method passes or fails the sanity check depends on the dataset and/or the model that is being used. This implies that the Parameter Randomization test should also be performed for each specific use case. This experimentally confirms the hypothesis posed by Yona and Greenfeld (2021) on natural image datasets.
- From the complementarity of results between coarse-grained and fine-grained attribution maps, we conclude that these methods might have to be evaluated in fundamentally different ways, focusing on single-pixel importance for fine-grained maps and on a more high-level view for coarse-grained maps. Further research is needed to verify or refute this hypothesis.
- Finally, we also introduce Seg-Sensitivity-n as an extension of Sensitivity-n (Ancona et al., 2018), and show that it has a higher signal-to-noise ratio than Sensitivity-n on high-dimensional datasets.

From these conclusions, we propose a set of benchmarking guidelines for practitioners seeking to select the best feature attribution method for their specific use case (see Fig. 9). We note that these guidelines should be viewed as exploratory, as more research is needed into which specific aspects of explanation methods are evaluated by the different metrics. This means that it is still difficult to prove that one method is strictly “better” than another, especially if metrics contradict each other. Our recommendation to practitioners is to use these benchmarking guidelines to perform a first selection of candidate methods, and then select one or multiple explanation methods based on use case-specific properties, such as computational budget, access to model internals, fine- or coarse-grainedness, and/or others.

- (1) *Baseline selection* First, a baseline attribution method must be defined. In general, a uniform random baseline can be used, but more specific baselines can also be chosen depending on the use case, such as an edge detector or some specific explanation method that one hopes to outperform.
- (2) *Metric selection* Next, a selection of metric implementations must be made. This can be done manually, if such a selection of metrics is obvious from the use case and there is a clear approach to masking available (2a in Fig. 9). In MNIST for example, masking using the black background color might be an intuitive choice. Alternatively, a *pilot study* can be performed (2b in Fig. 9). In such a pilot study, a large number of metrics and masking approaches are tested on a limited number of images. We then recommend computing inter-metric correlations and Krippendorff α values, and selecting those metric implementations that have high Krippendorff α and low inter-metric correlations. In this way, a minimal number of images can be used to draw dataset-wide conclusions, and metric scores will contain a minimal amount of redundant information.
- (3) *Parameter Randomization test* Before running a full benchmark on all available methods, we recommend first performing the Parameter Randomization test from AdEBAYO et al. (2018) using the spearman rank correlation. One should avoid using absolute values of attribution scores when performing this test, as this has been shown to unfairly penalize certain methods (Binder et al., 2023). Any methods that fail the test (by obtain-

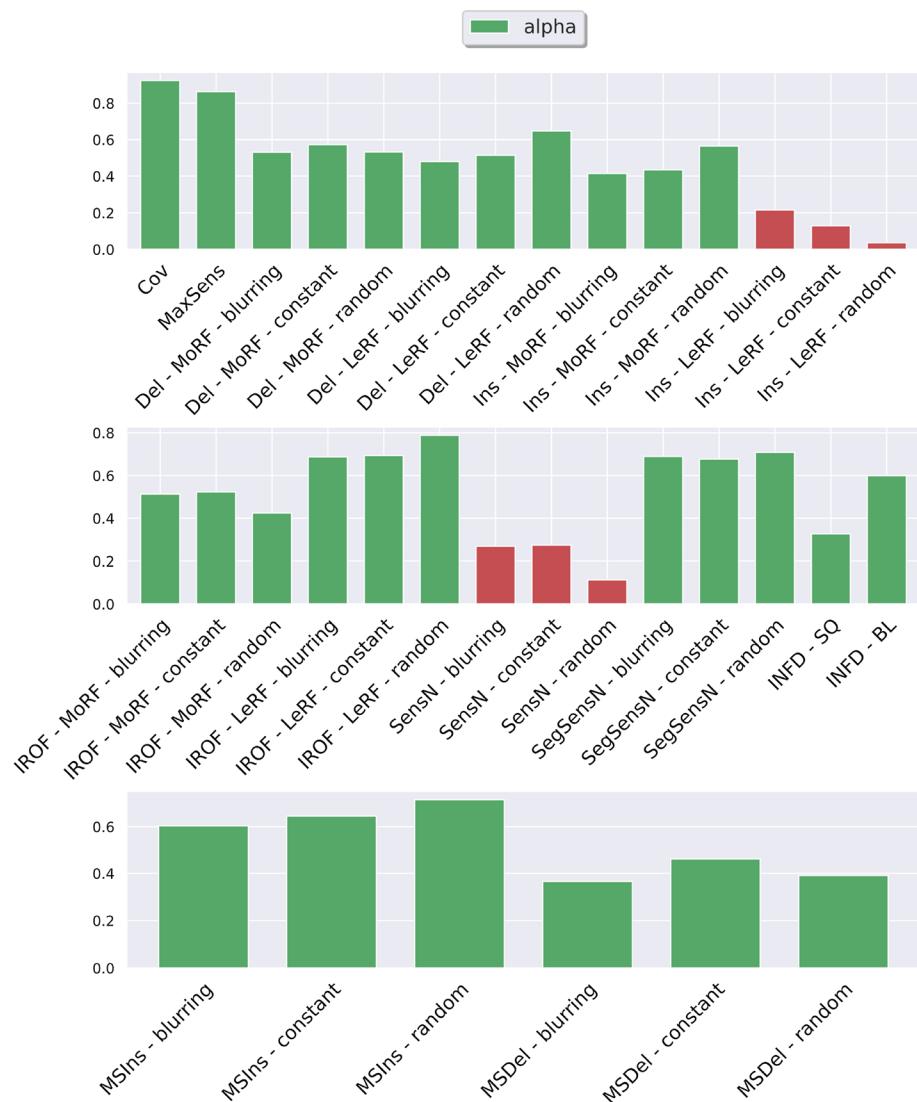


Fig. 15 Krippendorff α for all metrics. Values above 0.3 are shown in green, others are shown in red

ing a correlation score larger than a chosen threshold value) should be discarded from further analysis.

- (4) *Full benchmark* Once a selection of metrics is made and methods that fail the Parameter Randomization test are discarded, metric scores can be computed for the remaining methods on a large enough number of samples.
- (5) *Rough statistical analysis* Once the metric scores are computed for all methods and the baseline, a rough overview of method performance can be made using a paired t-test for each metric between each method and the random baseline. For those methods

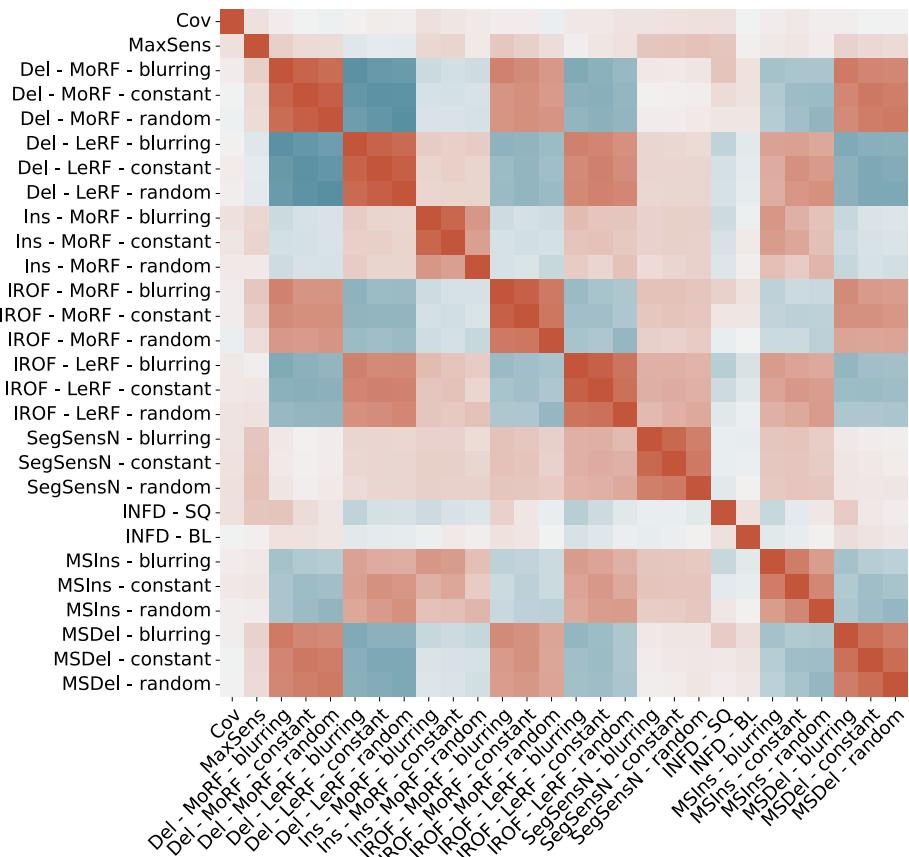


Fig. 16 Inter-metric correlations for all metrics with Krippendorff α larger than 0.3 (Color figure online)

that significantly outperform the baseline on a given metric, we recommend using the Cohen's d effect size to quantify performance.

(6) *Detailed comparison* Using the rough overview made in the previous step, we recommend selecting a smaller number of well-performing attribution methods, if possible with varying computational complexity. Those methods can then be compared in more detail using new paired t-tests, this time between the two methods rather than between a method and the baseline. We then recommend using the Probability of Superiority as an interpretable effect size measure to assess the fraction of cases where one method is superior to another. Based on these results, as well as other use-case specific constraints, a final selection of one or more ideal methods can then be made.

An example application of these guidelines can be found in Appendix B.

The results described in this paper leave a number of directions of future research. First of all, the observation that metrics do not necessarily measure the same underlying concept of feature attribution maps leads to the question of what those underlying concepts might be. A better understanding of those underlying concepts can lead to more directed benchmarking efforts and the development of better methods and/or

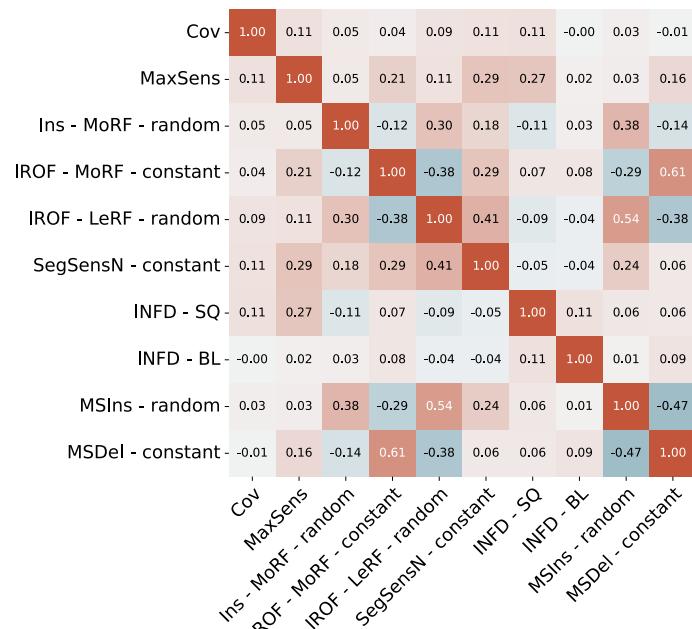


Fig. 17 Inter-metric correlations of final selection of metrics

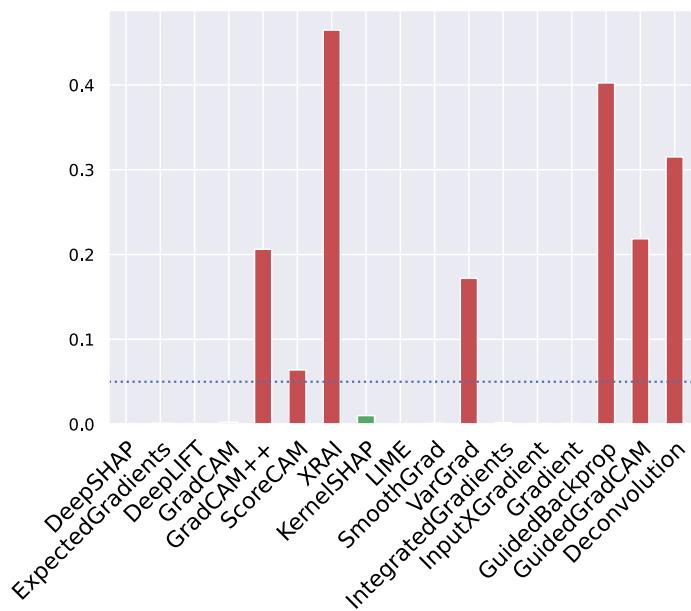


Fig. 18 Results of the Parameter Randomization test. All methods with $\rho > 0.05$ are discarded

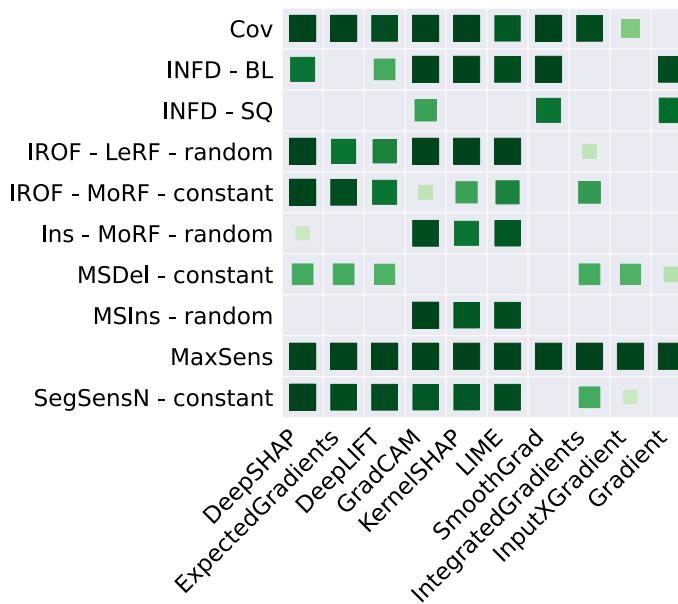


Fig. 19 Paired t-test results for selection of metrics

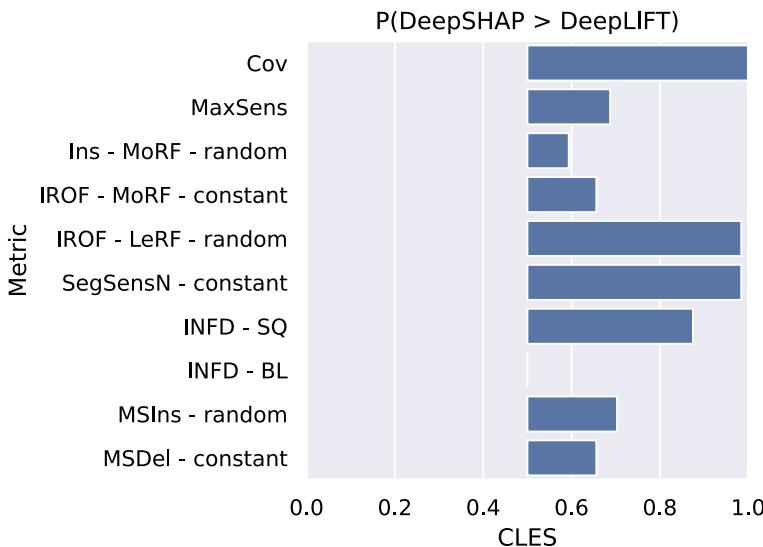


Fig. 20 Pairwise comparison between DeepSHAP and DeepLIFT

metrics. A possible link can be made with the concepts of necessity and sufficiency, found in the literature of causality (Pearl, 2009). Secondly, the complementarity of results for coarse- and fine-grained methods implies that a combination of different attribution maps can be more informative than a single one. This can lead to the development of new methods, generalizing the concept of feature attribution itself.

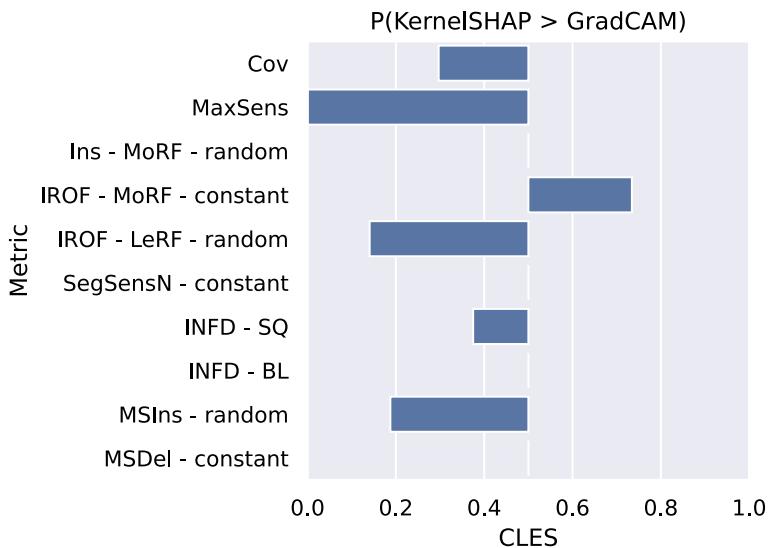
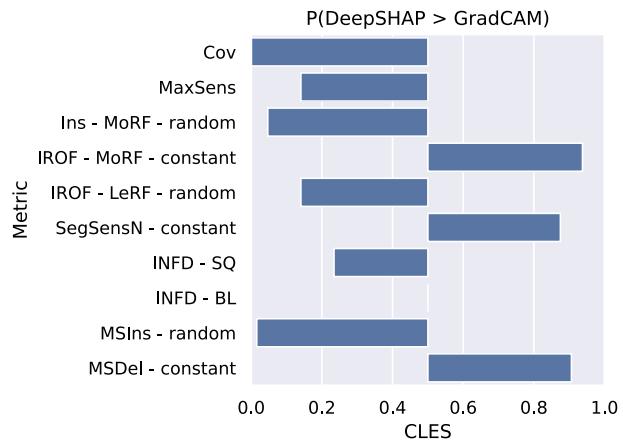


Fig. 21 Pairwise comparison between KernelSHAP and GradCAM

Fig. 22 Pairwise comparison between DeepSHAP and GradCAM



Future work can also be done in designing specific evaluation metrics for coarse- or fine-grained attribution methods. Finally, the application of the benchmark procedure on new datasets can shed light on what the best attribution methods are for a given problem domain. An important example is the domain of biomedical imaging. Here, medical practitioners are often interested in what the most important regions of a radiographic image are for a specific prediction (van der Velden et al., 2022), in order to build trust in the model and identify when a model might be making a mistake. Application of the general guidelines given above can help developers choose the right attribution method in this case.

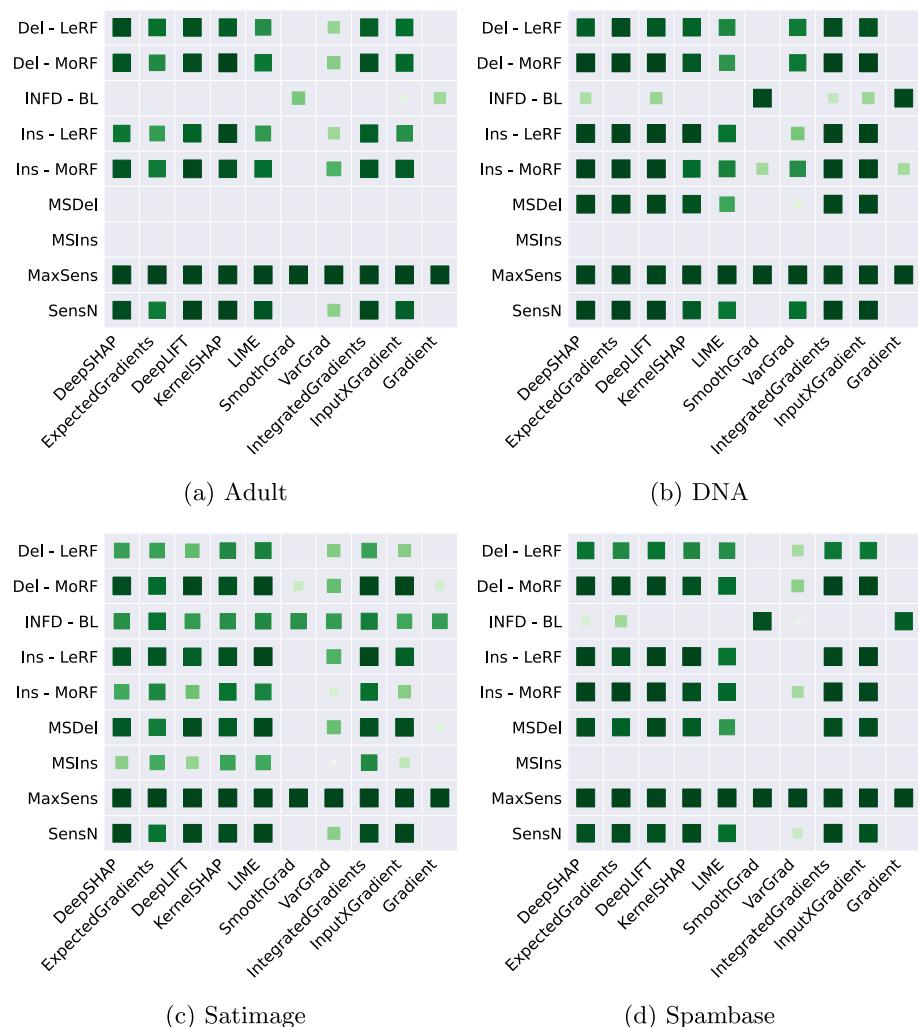


Fig. 23 Results of paired t-tests on tabular datasets

Appendix

A Supplementary figures

Full results of the paired t-tests for all metrics on all datasets are shown in Figs. 10, 11 and 12 for the low-, medium- and high-dimensional datasets, respectively. Krippendorff α values for all metrics on all datasets are shown in Fig. 13. Inter-metric correlations for all metrics on all datasets are shown in Fig. 14.

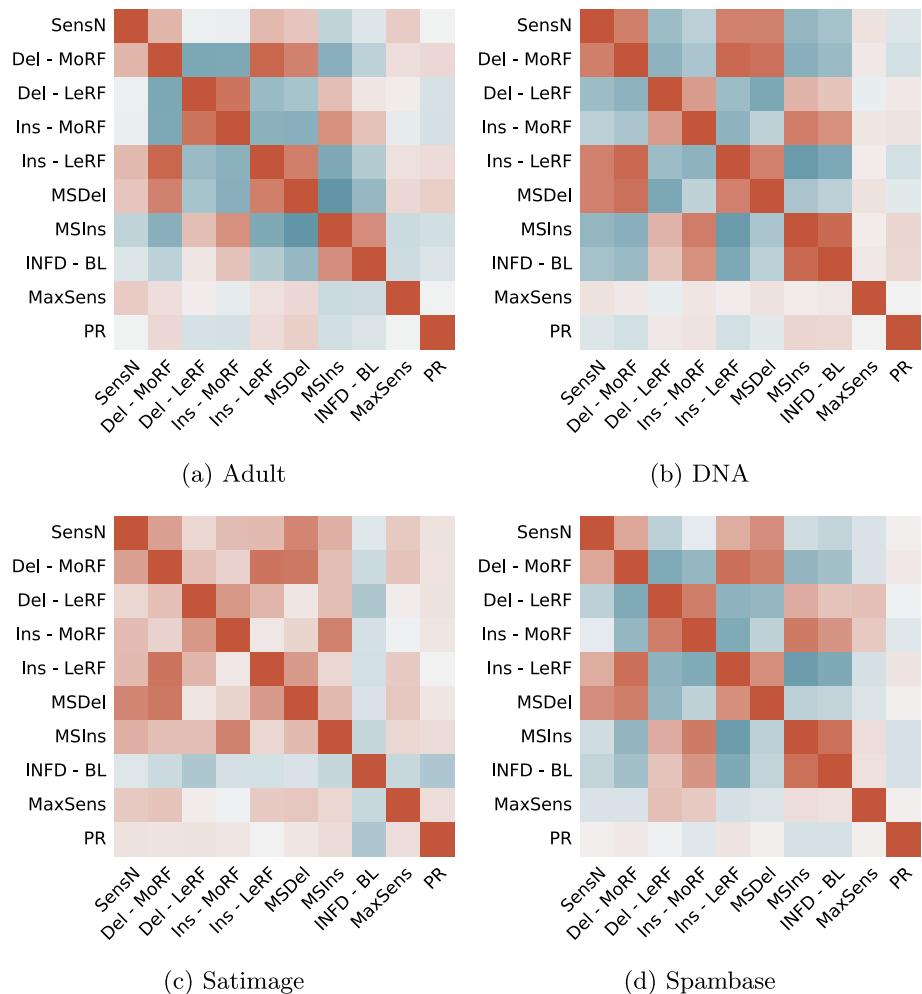


Fig. 24 Inter-metric correlations for tabular datasets

B Applying the guidelines to ImageNet

We apply the guidelines proposed in Sect. 8 to ImageNet for demonstration purposes:

- (1) *Baseline selection* Because we want to assume no prior knowledge about the problem setting, we select a uniform random baseline.
- (2) *Metric selection*
 - (a) We run all metrics using 64 samples as a small pilot study. The results from this pilot study are used to compute values for Krippendorff α . This is shown in Fig. 15. Metrics with $\alpha < 0.3$ are immediately discarded (shown in red in Fig. 15).

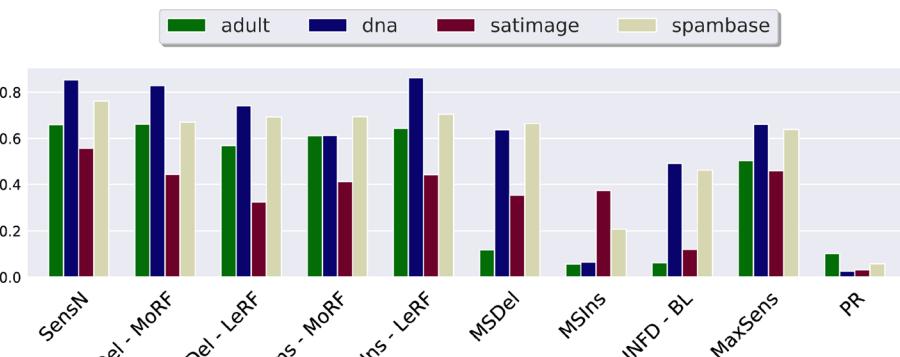


Fig. 25 Krippendorff α for tabular datasets

(b) We compute inter-metric correlations for the remaining metrics (shown in Fig. 16), and select metrics based on these correlations. The correlations for the final metric selection are shown in Fig. 17.

- (i) Different implementations of the same metric have very high correlations. For each metric type, we select the implementation with the highest Krippendorff α value.
- (ii) There are strong correlations between Deletion and IROF. Because we are working with high-dimensional data, we choose IROF over Deletion (Figs. 18 and 19).

(3) *Parameter Randomization test* Next, we perform the Parameter Randomization test from (Adebayo et al., 2018). We classify any method that obtains a Spearman rank correlation $\rho > 0.05$ as failing the test. All methods that fail the test are discarded.

(4) *Full benchmark* Once the metrics and methods are selected, the metric scores are computed on a larger number of images. We use 256 images for the full benchmark.

(5) *Rough statistical analysis* We perform paired t-tests on the results, comparing to the uniform random baseline. We observe two groups of methods with competitive but complementary results: the coarse-grained (CAM- and perturbation-based) methods, and DeepSHAP, DeepLIFT and ExpectedGradients.

(6) *Detailed comparison*

- (a) From the previous analysis, we select DeepSHAP, DeepLIFT, KernelSHAP and GradCAM for further analysis. DeepSHAP can be viewed as the computationally more expensive method in the fine-grained group, whereas DeepLIFT is computationally much cheaper. An analogous comparison can be made between KernelSHAP and GradCAM for the coarse-grained group.
- (b) We first compare DeepSHAP and DeepLIFT using the Probability of Superiority effect size. Results are shown in Fig. 20. We see that DeepSHAP significantly outperforms DeepLIFT on most metrics.
- (c) Next, we compare GradCAM and KernelSHAP. Results are shown in Fig. 21. Most of the paired t-tests between these two methods seem to be insignificant. Those metrics that do show a significant difference between the two methods

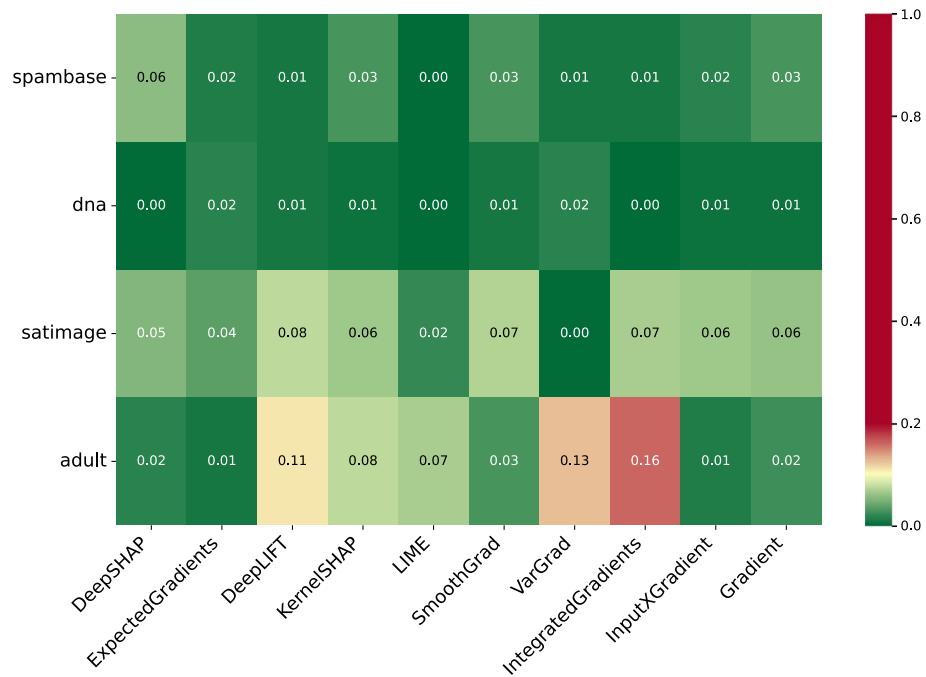


Fig. 26 Results of the Parameter Randomization metric on tabular datasets

disagree on which method is superior. Because no clear conclusion can be drawn in terms of the superiority of one method over the other, we select GradCAM for further analysis as it is computationally much cheaper than KernelSHAP.

(d) Finally, we compare DeepSHAP and GradCAM. Results for this comparison are shown in Fig. 22. We see that many of the paired t-test results are significant, but there is a strong complementarity in the results: some metrics favour DeepSHAP, others favour GradCAM. This suggests that both attribution maps might contain valuable complementary information. DeepSHAP is much more computationally expensive than GradCAM however, so depending on the use case, the developer might choose to provide both explanations, or to only use GradCAM explanations.

C Evaluating attributions on tabular datasets

To demonstrate the general applicability of the proposed methodology, we compute and evaluate feature attributions for tabular data. We use the Adult (Becker & Kohavi, 1996), DNA (King, 1992), Satimage (Srinivasan, 1993) and Spambase (Hopkins et al., 1999) datasets from the OpenML repository (Vanschoren et al., 2013).

On each of these datasets, a fully-connected neural network with two hidden layers of 64 neurons each is trained. Because this model has no convolutional layers, only the

methods that require differentiability of the model or that have no model requirements at all were evaluated (see Table 2). We evaluated all metrics that are applicable to any type of data (see Table 1). Masking was done by replacing feature values by 0, which after standard scaling of the features is equivalent to masking using the feature mean value.

C.1 Paired t-tests

Results of the paired t-tests are shown in Fig. 23. We see that the results are very similar across the four datasets, with the exception of the minimal subset metrics. This might be linked to the number of classes in the dataset, which is 2 for the Adult and Spambase datasets, 3 for the DNA dataset and 6 for the Satimage dataset.

If the dataset has only 2 classes, a constant 0 vector has a 50% probability of producing the same output as the original sample (assuming no class imbalance). Combined with the fact that these datasets have much fewer features than image datasets, there is a reasonable probability that masking any number of features for a given sample will not flip the result. In this case, both minimal subset insertion and minimal subset deletion will have a constant score for all methods. This might explain why the minimal subset metrics have much higher p-values for the Adult and Spambase datasets than for the DNA and Satimage datasets. More experiments can be done to verify this hypothesis.

C.2 Inter-metric correlations

Inter-metric correlations for the tabular datasets are shown in Fig. 24. We see that the correlations vary strongly across the four datasets. This observation confirms the need for a separate benchmarking experiment for each dataset, as proposed in Sect. 8.

C.3 Ranking consistency

Ranking consistency for the tabular datasets is visualized in Fig. 25. We see that the ranking consistency is generally a bit higher than for the image datasets, but there is again a large variation across datasets.

C.4 Parameter randomization

Results of the Parameter Randomization test are shown in Fig. 26. We see that the absolute rank correlation is very low in most cases, with the exception of the Adult dataset. Although the absolute rank correlation remains below 0.25, it is notably higher than for the other datasets. This might be explained by the fact that the Adult dataset has the fewest features of all four datasets, which makes it easier for a random permutation of the features to have a higher rank correlation with the original ordering. Further experiments, where the rank correlation is corrected for the total number of features, can be done to verify or refute this hypothesis.

Author contributions AG and AJR have jointly implemented the different attribution methods and metrics used in the study. AG has then performed the experiments and statistical analysis of the results. TB, DV, TDB and YS have supervised the project and assisted in writing the paper.

Funding The research leading to these results has received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme, and from the BOF project 01D13919.

Data availability The necessary code to reproduce the experiments is available at <https://github.com/arngevaert/benchmark-general-imaging>. The implementation of attribution metrics is available at <https://github.com/arngevaert/benchmark>. Datasets, results of the experiments and model parameters are available at <https://zenodo.org/record/6221586#.Yos8IHVByV4> and <https://zenodo.org/record/6205531#.Yos8H3VByV4>, respectively.

Code availability The Python code associated with this work is available at <https://github.com/arngevaert/benchmark-general-imaging>.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Suesstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274–2281. <https://doi.org/10.1109/TPAMI.2012.120>

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., & Kim, B. (2018). Sanity checks for saliency maps. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems*. (Vol. 31). UK: Curran Associates, Inc.

Ancona, M., Ceolini, E., Öztïreli, C., & Gross, M. (2018). Towards better understanding of gradient-based attribution methods for deep neural networks. International conference on learning representations. Retrieved from <https://openreview.net/forum?id=Sy21R9JAW>

Balduzzi, D., Frean, M., Leary, L., Lewis, J. P., Ma, K. W.-D., & McWilliams, B. (2017). The shattered gradients problem: If resnets are the answer, then what is the question? In D. Precup & Y. Teh (Eds.), International conference on machine learning (Vol. 70).

Becker, B., & Kohavi, R. (1996). Adult. UCI Machine Learning Repository. <https://doi.org/10.24432/C5XW20>

Binder, A., Weber, L., Lapuschkin, S., Montavon, G., Müller, K.-R., & Samek, W. (2023). Shortcomings of top-down randomization-based sanity checks for evaluations of deep neural network explanations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (p. 16143–16152).

Bland, J. M., & Altman, D. G. (1995). Multiple significance tests: The Bonferroni method. *BMJ*, 310(6973), 170. <https://doi.org/10.1136/bmj.310.6973.170>

Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., & Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 1877–1901). UK: Curran Associates, Inc.

Chandrashekhar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), 16–28. <https://doi.org/10.1016/j.compeleceng.2013.11.024>

Chattopadhyay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)* (pp. 839–847). <https://doi.org/10.1109/WACV.2018.00097>

Chen, H., Janizek, J. D., Lundberg, S., & Lee, S.-I. (2020). True to the model or true to the data?

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). UK: Routledge. <https://doi.org/10.4324/9780203771587>

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (cvpr'05)* (Vol. 1, pp. 886–893). <https://doi.org/10.1109/CVPR.2005.177>

Dandl, S., Molnar, C., Binder, M., & Bischl, B., et al. (2020). Multi-objective counterfactual explanations. In T. Bäck (Ed.), *Parallel problem solving from nature–ppsn* (pp. 448–469). UK: Springer International Publishing.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, H., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA (pp. 248–255). <https://doi.org/10.1109/CVPR.2009.5206848>

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning.

Erion, G., Janizek, J. D., Sturmels, P., Lundberg, S. M., & Lee, S.-I. (2021). Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, 3(7), 620–631. <https://doi.org/10.1038/s42256-021-00343-w>

Fernandez, F.-G. (2020). Torchcam: Class activation explorer. <https://github.com/frgfm/torch-cam>. GitHub.

Fong, R.C., & Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 3449–3457). <https://doi.org/10.1109/ICCV.2017.371>

Ghorbani, A., Abid, A., & Zou, J. (2019). Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence* (vol. 33(01), pp. 3681–3688). Retrieved from <https://ojs.aaai.org/index.php/AAAI/article/view/4252> <https://doi.org/10.1609/aaai.v33i01.33013681>

Griffin, G., Holuband, A., & Perona, P. (2022). Caltech 256. CaltechDATA. <https://doi.org/10.22002/D1.20087>

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770–778). <https://doi.org/10.1109/CVPR.2016.90>

Hedström, A., Weber, L., Lapuschkin, S., & Höhne, M. (2023). Sanity checks revisited: An exploration to repair the model parameter randomisation test. In *Xai in action: Past, present, and future applications*. Retrieved from <https://openreview.net/forum?id=vVpefYmnnsG>

Hooker, S., Erhan, D., Kindermans, P.-J., & Kim, B. (2019). A benchmark for interpretability methods in deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*. (Vol. 32). UK: Curran Associates, Inc.

Hopkins, M., Reeber, E., Forman, G., & Suermondt, J. (1999). Spambase. UCI Machine Learning Repository. <https://doi.org/10.24432/C53G6X>

Kapishnikov, A., Bolukbasi, T., Viegas, F., & Terry, M. (2019). Xrai: Better attributions through regions. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*.

King, R. (1992). Molecular biology (splice-junction gene sequences). UCI Machine Learning Repository. <https://doi.org/10.24432/C5M888>

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., & Reblitz-Richardson, O. (2020). Captum: A unified and generic model interpretability library for pytorch.

Krippendorff, K. (2011). Computing krippendorff's alpha-reliability. Citeseer.

Krippendorff, K. (2019). *Content analysis: An introduction to its methodology*. UK: SAGE Publications, Inc. <https://doi.org/10.4135/9781071878781>

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, & K. Weinberger (Eds.), *Advances in neural information processing systems*. (Vol. 25). UK: Curran Associates, Inc.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE* (Vol. 86, pp. 2278–2324). <https://doi.org/10.1109/5.726791>

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2019). Continuous control with deep reinforcement learning.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision—ECCV 2014* (pp. 740–755). UK: Springer International Publishing.

Lin, Z. Q., Shafee, M. J., Bochkarev, S., Jules, M. S., Wang, X. Y., & Wong, A. (2019). Do explanations reflect decisions? a machine-centric strategy to quantify the performance of explainability algorithms.

Liu, X., Wang, X., & Matwin, S. (2018). Improving the interpretability of deep neural networks with knowledge distillation. In *2018 IEEE international conference on data mining workshops (ICDMW)* (pp. 905–912). <https://doi.org/10.1109/ICDMW.2018.00132>

Lundberg, S. M., & Lee, S.-I., et al. (2017). A unified approach to interpreting model predictions. In I. Guyon (Ed.), *Advances in neural information processing systems*. (Vol. 30). UK: Curran Associates, Inc.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. <https://doi.org/10.1038/nature14236>

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Pearl, J. (2009). *Causality* (2nd ed.). UK: Cambridge University Press.

Petsiuk, V., Das, A., & Saenko, K. (2018). Rise: Randomized input sampling for explanation of black-box models. In *British machine vision conference (BMVC)*. Retrieved from <http://bmvc2018.org/contents/papers/1064.pdf>

Ribeiro, M.T., Singh, S., & Guestrin, C. (2016). “why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939778>

Ribeiro, M.T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, 32(1). <https://doi.org/10.1609/aaai.v32i1.11491>

Rieger, L., & Hansen, L. (2020). Irof: A low resource evaluation metric for explanation methods. In *Proceedings of the workshop AI for affordable healthcare at ICLR 2020*.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., & Müller, K.-R. (2017). Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11), 2660–2673. <https://doi.org/10.1109/TNNLS.2016.2599820>

Schmidt, P., & Biessmann, F. (2019). Quantifying interpretability and trust in machine learning systems. In *Aaaai 2019 workshop on network interpretability for deep learning*.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradientbased localization. In *Proceedings of the IEEE international conference on computer vision (ICCV)*.

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning* (Vol. 70, pp. 3145–3153). PMLR.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676), 354–359. <https://doi.org/10.1038/nature24270>

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In Y. Bengio & Y. LeCun (Eds.), *2nd international conference on learning representations, ICLR 2014*, Workshop track proceedings.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015*, Conference track proceedings.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., & Wattenberg, M. (2017). Smoothgrad: Removing noise by adding noise.

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for Simplicity: The All Convolutional Net. In *International conference on learning representations—workshop track proceedings*, 3. [arXiv:1412.6806](https://arxiv.org/abs/1412.6806)

Srinivasan, A. (1993). Statlog (Landsat Satellite). UCI Machine Learning Repository. <https://doi.org/10.24432/C55887>

Sturmels, P., Lundberg, S., & Lee, S.-I. (2020). Visualizing the impact of feature attribution baselines. *Distill*. (<https://distill.pub/2020/attributionbaselines>) <https://doi.org/10.23915/distill.00022>

Sundararajan, M., & Najmi, A. (2020). The many shapley values for model explanation. H.D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (Vol. 119, pp. 9269–9278). PMLR.

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In D. Precup & Y.W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning* (Vol. 70, pp. 3319–3328). PMLR.

Tomsett, R., Harborne, D., Chakraborty, S., Gurram, P., & Preece, A. (2020). Sanity checks for saliency metrics. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), 6021–6029. <https://doi.org/10.1609/aaai.v34i04.6064>

van der Velden, B. H., Kuijff, H. J., Gilhuijs, K. G., & Viergever, M. A. (2022). Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis*, 79, 102470. <https://doi.org/10.1016/j.media.2022.102470>

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2), 49–60. <https://doi.org/10.1145/2641190.2641198>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I., et al. (2017). Attention is all you need. In I. Guyon (Ed.), *Advances in neural information processing systems*. (Vol. 30). UK: Curran Associates, Inc.

Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology (Harvard JOLT)*, 31, 841.

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., & Hu, X. (2020). Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops*.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612. <https://doi.org/10.1109/TIP.2003.819861>

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. [arXiv:1708.07747](https://arxiv.org/abs/1708.07747)

Yang, M., & Kim, B. (2019). Benchmarking attribution methods with relative feature importance.

Yeh, C.-K., Hsieh, C.-Y., Suggala, A., Inouye, D. I., & Ravikumar, P. K. (2019). On the (in) fidelity and sensitivity of explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. D' alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*. (Vol. 32). UK: Curran Associates, Inc.

Yona, G., & Greenfeld, D. (2021). Revisiting sanity checks for saliency maps. In *Explainable AI approaches for debugging and diagnosis*.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision–ECCV 2014* (pp. 818–833). UK: Springer International Publishing.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: A 10 million image database for scene recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2017.2723009>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Arne Gevaert¹  · Axel-Jan Rousseau² · Thijs Becker⁴ · Dirk Valkenborg² · Tijl De Bie³ · Yvan Saeys¹

✉ Arne Gevaert
arne.gevaert@ugent.be

Axel-Jan Rousseau
axeljan.rousseau@uhasselt.be

Thijs Becker
thijs.becker@vito.be

Dirk Valkenborg
dirk.valkenborg@uhasselt.be

Tijl De Bie
tijl.debie@ugent.be

Yvan Saeys
yvan.saeys@ugent.be

¹ Department of Applied Mathematics, Computer Science and Statistics: Data mining and Modeling for Biomedicine (DaMBi), Ghent University, Technologiepark-Zwijnaarde 71, 9052 Ghent, Belgium

² Center for Statistics (CENSTAT), Hasselt University, Agoralaan Building D, 3590 Diepenbeek, Belgium

³ IDLab, Department of Electronics and Information Systems, Ghent University, Technologiepark-Zwijnaarde 19, 9052 Ghent, Belgium

⁴ VITO, Boeretang 200, 2400 Mol, Belgium