Using the Path of Least Resistance to Explain Deep Networks

Anonymous Authors[1]

Abstract

Integrated Gradients (IG), a widely used path-based attribution method, assigns importance scores to input features by integrating gradients of the models along a straight path from a baseline to the input. While effective in certain cases, we show that choosing straight paths can lead to flawed attributions. In this paper, we identify how these misattributions arise. As a solution, we propose a new approach that considers the input space as a Riemannian manifold and computes attributions by integrating gradients of the model along geodesics. We call our approach *Geodesic Integrated Gradients*. We present an approximation method to make the computations of Geodesic IG seamless. Furthermore, in our experiments with both synthetic and real world data, we demonstrate that our approach outperforms existing methods including the original IG.

1. Introduction

The use of deep learning models has risen in many applications. With it, so too has the desire to understand why these models make certain predictions. These models are often referred to as "opaque", as it is difficult to discern the reasoning behind their predictions (Marcus, 2018). Additionally, deep learning models can inadvertently learn and perpetuate biases found in their training data (Sap et al., 2019). To create fair and trustworthy algorithms, it is essential to be able to explain a model's output (Das & Rad, 2020).

Some of the methods proposed to explain neural networks include DeepLIFT (Shrikumar et al., 2017), Layer-wise Relevance Propagation (LRP) (Bach et al., 2015) and Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016). For a summary of the most recent explanations see Holzinger et al. (2022).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Significant effort has been dedicated to designing explanation methods that satisfy certain desirable axioms. This is due to the lack of ground truth for evaluating them. The axioms can ensure that the explanations are principled. One of the most successful axiomatic methods is Integrated Gradients (IG) (Sundararajan et al., 2017). Consider a function $f : R^n \to R$, representing the neural network and an input vector $\mathbf{x} \in R^n$. Furthermore, consider a baseline input vector $\overline{\mathbf{x}} \in R^n$ (typically chosen such that the network gives baseline a near zero score). IG explains the network by quantifying how much of the difference $f(\mathbf{x}) - f(\overline{\mathbf{x}})$ can be attributed to the ith dimension of $\mathbf{x}$, $\mathbf{x}_i$.

Integrated Gradient gives attribution IG_i to the ith dimension of the input by solving the following path integral

$$IG_i(\mathbf{x}) = (\mathbf{x}_i - \overline{\mathbf{x}}_i) \int_0^1 \frac{\partial f(\gamma(t))}{\partial \mathbf{x}_i} dt, \tag{1}$$

where $\gamma(t) = \overline{\mathbf{x}} + t(\mathbf{x} - \overline{\mathbf{x}})$ is a straight path from the baseline to input. The claim of the creators of IG is that Eq. 1 tells us how the model got from predicting essentially nothing at $\overline{\mathbf{x}}$ to giving the prediction at $\mathbf{x}$. Considering gradients represent the rate of change of functions, the above expression should tell us how scaling each feature along the path affects the increase in the network score for the predicted class.

Nevertheless, in this paper we show that defining such an attribution along a straight path on Euclidean space can lead to misattributions. We introduce what we call **Geodesic Integrated Gradients**, which generalises the above setting to geodesic paths on a Riemannian manifolds, circumventing the above pitfalls, whilst still adhering to all the axioms of IG.

Before making the case for our Geodesic Integrated Gradient, let us first show an example of an artefact that can arise from choosing straight paths, generating explanations which do not reflect the true behaviour of a model.

We highlight this issue on a half-moons classification task. We train a simple multi-layer perceptron (MLP) with 3 layers, ReLU activations and a cross-entropy loss to distinguish the upper moon from the lower one. The cross-entropy is split into a final log-softmax activation and a negative log-likelihood loss, so that we can explain probabilities. We show an example of the half-moons data in 1, with the model's gradients field illustrated with gray arrows.

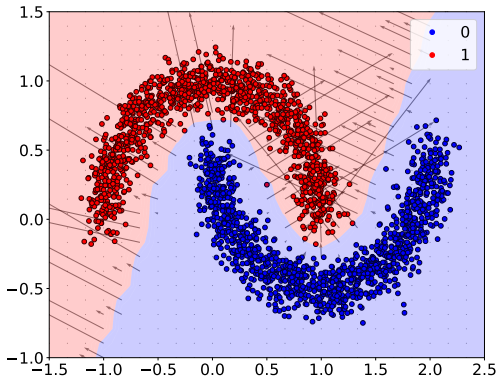

Figure 1. **Half moons dataset** with Gaussian noise $\mathcal{N}(0, 0.1)$. Predictions of an MLP are illustrated in blue and red colours. The gradient field of the model is represented by gray arrows. We can see that, as expected, the gradients are very high right on the decision boundary. However, they drop off to nearly zero everywhere else. The regions on each side of the decision boundary are shaded with different colours. The accuracy on the test set for this model is 99.9%.

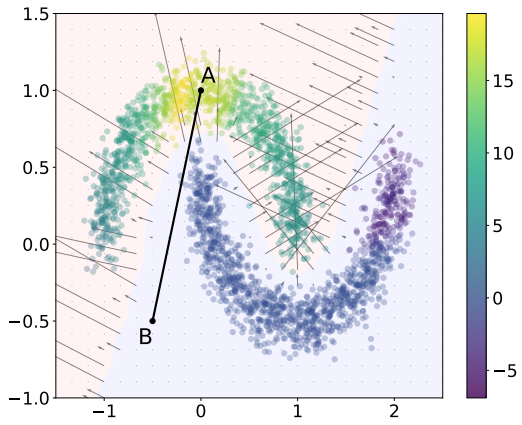

Figure 2. **Integrated Gradients attributions.** The colour map represents $\text{IG}_1(\mathbf{x})$ for each point $\mathbf{x}$, with $(-0.5, -0.5)$ (point B) as the baseline. Points around A have much higher attributions than other points on the top moon, despite all being sufficiently above the decision boundary of the MLP. This is due to the path being close to the decision boundary, resulting in high gradients (gray arrows) along this path.

We now compute Integrated Gradients, Eq. 1, for this model on the test data. Let us consider a baseline and input pair, such that the baseline is outside of either half-moon, for example at (-0.5, -0.5). This is a good choice of baseline, since network should assign near-zero score to it. Let us call the feature in the vertical axis the 1st component of x. In Fig. 2 we illustrate the attribution of this feature, $IG_1(\mathbf{x})$, for each point using the colour map. One should expect to see all the points sufficiently above the decision boundary to receive equally high attributions. Intuitively, this is expected, because if a point is above the decision boundary, its x_1 component is an important factor in the classification. However, for a model that is very skillful at the classification task, since the point is significantly above the decision boundary, going slightly down should not make any difference. Because in such a model's score should not change significantly anywhere other than near the decision boundary. However, we can see in Fig. 2 that some points on the upper moon receive much higher x_1-attribution than others. These are points such that a straight line from the baseline to them mostly falls on high gradient regions. This does not reflect the model's behaviour. A similar point could be made about the horizontal axis. This is in contrast with Fig. 3, where we show our method gives equally high attribution to all points sufficiently above the decision boundary, with different shades for the points closer to the boundary in x_1 direction. In Section 3.1 we present the details of how our method that achieves the results presented in this figure.

Before giving the formal method in section 2, let us discuss the intuition behind our Geodesic IG. We want the path in Eq. 1 to be such that it avoids regions with high model gradient. This is because failing to do so would superficially increase the result of the integral, leading to the types of artefacts illustrated in Fig. 2. Therefore, we should try to find the path of least resistance, as this is the path that avoids steep gradients as much as possible. As we shall see in section 2, the input space can be viewed as a Riemannian manifold with a metric derived from the model gradients. The path of least resistance between a chosen baseline and input, therefore, is the geodesic path between the two points.

In section 2, we also develop a method to approximate the geodesic path between two points on the manifold to make solving the problem computationally feasible. We then show that geodesic IG satisfies all the axioms of IG, including symmetry. The symmetry axiom is defined in the following way.

Definition 1.1. Consider an input-baseline pair $\mathbf{x}$ and $\bar{\mathbf{x}}$, and a function f that is symmetric in dimensions i and j. If $\mathbf{x}_i = \mathbf{x}_j$ and $\bar{\mathbf{x}}_i = \bar{\mathbf{x}}_j$, then an attribution method is Symmetry-Preserving if $attr_i(\mathbf{x}; f) = attr_j(\mathbf{x}; f)$, where $attr_n(\mathbf{x}; f)$ is the attribution of $\mathbf{x}_n$.

(Sundararajan et al., 2017, Theorem 1) shows that IG is the only path method that satisfies symmetry on Euclidean space. We generalise this theorem for Geodesic IG on Riemannian manifolds.

In Section 3, we demonstrate the effectiveness of the Geodesic IG method on the real-world Pascal VOC 2012 dataset (Everingham et al.). Our results outperform existing

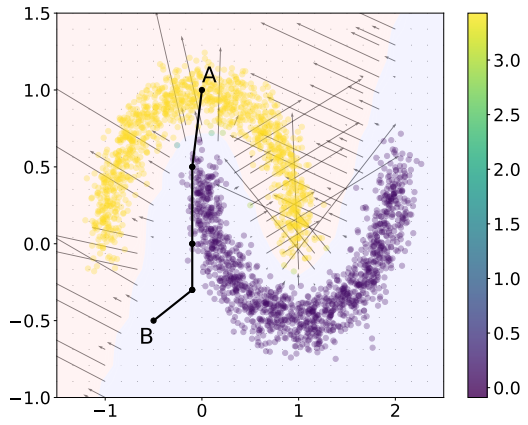

Figure 3. **Geodesic IG attributions.** Geodesic IG successfully avoids high gradients regions, and presents as a result attributions free of any artefacts. Moreover, unlike IG, there is no high variation of attributions between close points. We provide a rigorous comparison of Geodesic IG against various baselines in the Experiment section.

methods, as we evaluate using various metrics. We also provide supplementary experiments and and ablation studies in the appendix.

Section 4 reviews related work, including the comparison of Geodesic IG with other methods that attempt to overcome the shortcomings of Integrated Gradients.

2. Method

In section 1, we gave the intuition that using geodesic paths can correct the misattribuion in IG that arise from integrating along straight paths. Let us now formalise this idea.

Geodesic distance formulation. Let us define a neural network as a function $f : \mathbb{R}^n \to \mathbb{R}$, where n is the dimension of the input space. Let us also define $\mathbf{x}$ a point in this input space. We denote the Jacobian of f at $\mathbf{x}$ as $\mathrm{J}_\mathbf{x}$.

Using Taylor's theorem, for a vector $\boldsymbol{\delta}$ with an infinitesimal norm: $\forall \epsilon, ||\boldsymbol{\delta}|| \leq \epsilon$, we have:

$$||f(\mathbf{x}+\boldsymbol{\delta}) - f(\mathbf{x})|| \approx ||\mathrm{J}_\mathbf{x}\boldsymbol{\delta}|| \approx \boldsymbol{\delta}^T \mathrm{J}_\mathbf{x}^T \mathrm{J}_\mathbf{x} \boldsymbol{\delta} \tag{2}$$

Using equation 2, we can now define a tangent space $\mathrm{T}_\mathbf{x}\mathrm{M}$ of all $\boldsymbol{\delta}$, equipped with a local inner product $\mathrm{G}_\mathbf{x}$:

$$< \boldsymbol{\delta}, \boldsymbol{\delta}' >_\mathbf{x} = \boldsymbol{\delta}^T \mathrm{G}_\mathbf{x} \boldsymbol{\delta}' = \boldsymbol{\delta}^T \mathrm{J}_\mathbf{x}^T \mathrm{J}_\mathbf{x} \boldsymbol{\delta}' \tag{3}$$

As a result, we can view the input space as a Riemannian manifold $(\mathbb{R}^n, \mathrm{G})$, where the Riemannian metric G is defined above. On this manifold, the length of a curve $\gamma(t) : [0,1] \to \mathbb{R}^n$ is defined as:

$$\begin{aligned} \mathrm{L}(\gamma) &= \int_0^1 \sqrt{< \dot\gamma(t), \dot\gamma(t) >_{\gamma(t)}} dt \\ &= \int_0^1 ||\partial_t f(\gamma(t)) \times \dot\gamma(t)||\, dt, \end{aligned} \tag{4}$$

where $\dot\gamma(t)$ is the derivative of $\gamma(t)$ with respect to t. The **geodesic distance**, denoted L^*, between **a** and **b** is then defined as the minimum length among curves γ such that $\gamma(0) = \mathbf{a}$ and $\gamma(1) = \mathbf{b}$. We also call **geodesic path** the curve γ^* which minimises the length L. This path can be interpreted as the shortest path between **a** and **b** in the manifold.

Remark 2.1. We can infer from Equation 4 that the geodesic path avoids as much as possible high-gradients regions. This is the main desired property of a path to be used for path-based attributions. Representing the path of least resistance, the geodesic path circumvents superficially high values of attributions.

Approximation of the geodesic. Computing the exact geodesic would require computing L on an infinite number of paths γ, which is not possible in practice. However, several methods have been proposed to approximate this value. We draw from previous work (Yang et al., 2018; Chen et al., 2019) and present one with desirable characteristics.

First, we compute the K Nearest Neighbors (kNN) algorithm on points between (and including) input and baseline. These points can be either sampled or generated. The geodesic distance between two neighbouring points, $\mathbf{x}_i$ and $\mathbf{x}_j$, can be approximated by a straight path $\mathbf{x}_i + t \times (\mathbf{x}_j - \mathbf{x}_i)$. We have the above approximation because for dense enough data, the euclidean distance between neighbouring points is a good approximation of the geodesic distance. This reflects the fact that a small region of a Riemannian manifold, called Riemann neighbourhood, is locally isometric to a Euclidean space[1]. So the geodesic distance between the two neighbouring points is approximated by:

$$\begin{aligned} \mathrm{L}^*_{ij} &= \int_0^1 ||\partial_t f(\mathbf{x}_i + t \times (\mathbf{x}_j - \mathbf{x}_i)) \times (\mathbf{x}_i - \mathbf{x}_j)||\, dt \\ &= ||\mathbf{x}_i - \mathbf{x}_j|| \int_0^1 ||\partial_t f(\mathbf{x}_i + t \times (\mathbf{x}_j - \mathbf{x}_i))||\, dt \end{aligned} \tag{5}$$

Equation 5 corresponds to the original Integrated Gradients method, albeit with the norm. This integral can be approx-

[1]We shall further formalise this intuition later in this section.

imated by a Riemannian sum similarly to (Sundararajan et al., 2017):

$$\mathrm{L}^*_{ij} \approx ||\mathbf{x}_i - \mathbf{x}_j|| \sum_{k=0}^{m} ||\partial f(\mathbf{x}_i + \frac{k}{m} \times (\mathbf{x}_j - \mathbf{x}_i))|| \quad (6)$$

For input-baseline pair, $\mathbf{x}$ and $\overline{\mathbf{x}}$, we can now see the set ($\mathbf{x}$, $\overline{\mathbf{x}}$, $\mathbf{x}_i$) as a weighted graph, with the weights being the geodesic distances between two neighbors L^*_{ij}. To compute the geodesic path between $\mathbf{x}$ and $\overline{\mathbf{x}}$, we can use a shortest path algorithm, such as Dijkstra or A* with the euclidean distance as the heuristic.

The resulting Geodesic Integrated Gradients corresponds to the sum of the gradients along this shortest path:

$$\text{Geodesic IG}_i(\mathbf{x}) = (x_i - \overline{x}_i) \sum_{k=0}^{m} \int_0^1 \frac{\partial f(\mathbf{x}^k + t \times (\mathbf{x}^{k+1} - \mathbf{x}_k))}{x_i^k} \, dt \quad (7)$$

where $\mathbf{x}^k$ are the points along the shortest path. The integrals in Equation 7 can also be approximated with Riemannian sums.

The gradients between each pair of neighbours can also be estimated in batches to speed up the attribution computation. Moreover, several inputs' attributions can be computed together, with similar speed as IG: if we want to compute the attribution of N inputs, with 10 interpolation steps and 5 nearest neighbors, the number of gradients to calculate is $10 \times 5 \times \mathrm{N} = 50\mathrm{N}$, which amounts to computing IG with 50 steps. This does not include the computation of the shortest path, which is for instance $O(\mathrm{N}^2)$ for Dijkstra algorithm. Please also refer to Figure 4 for an illustration of this method.

Symmetry preserving of Geodesic IG The family of generalisations of Integrated Gradients to non-straight paths, such as Eq. 7, is called *path methods* of explanation. We see in Sundararajan et al. (2017) that all path methods satisfy all of the axioms that IG is based on, apart from the symmetry axiom that we discussed in the introduction. Sundararajan et al. (2017, Theorem 1) shows that Integrated Gradients is the only path-method on Euclidean surfaces that is symmetry preserving. Here we demonstrate that Geodesic Integrated Gradients satisfies symmetry property on Riemannian manifolds, and it is the only path-based method that does so.

Let the ith and jth dimensions of $\gamma(t)$ be $\gamma_i(t)$ and $\gamma_j(t)$ respectively and f be a function differentiable almost everywhere on t. Furthermore, take f to be symmetric with respect to x_i and x_j. If $\gamma_i(t) = \gamma_j(t)$ for all $t \in [0, 1]$, then we have

$$||\partial_t f(\gamma_i(t)) \times \dot{\gamma}_i(t)|| = ||\partial_t f(\gamma_j(t)) \times \dot{\gamma}_j(t)||, \quad (8)$$

almost everywhere on t. Therefore, the ith and jth components of Eq. 4 are equal. Furthermore, since Eq. 7 integrates along the path that is an approximation of Eq. 4, we have Geodesic IG_i = Geodesic IG_j. Indeed our geodesic paths satisfy $\gamma_i(t) = \gamma_j(t)$ for all $t \in [0, 1]$ on the Riemannian manifolds. To see this, let us select a baseline $\overline{\mathbf{x}}$ and U a Riemann neighbourhood centred at $\overline{\mathbf{x}}$. Let us also define the geodesic path γ such as $\gamma(0) = \overline{\mathbf{x}}$. Further, define $\mathbf{v}(t) := \gamma'(t)$, where γ' is the derivative of γ. Then, in the local coordinates system of the neighbourhood of any point, called normal coordinates, we have $\gamma(t) = (tv_1(t), ..., tv_n(t))$. Since the function is symmetric in the ith and jth dimensions, we have v_i and v_j are the same everywhere. From this, we can see that $\gamma_i(t) = \gamma_j(t)$ for all $t \in [0, 1]$ and therefore Geodesic IG satisfies symmetry. In other words, since the geodesic generalises straight paths to Riemannian manifolds, it follows that the symmetry property of IG on Euclidean space is extended to Geodesic IG on Riemannian manifolds. In fact, using the same argument as above, the proof of Sundararajan et al. (2017, Theorem 1) can be readily used to show that Geodesic IG is the only path method on Riemannian manifolds that satisfy symmetry.

Assumption of the approximation. Here we formalise the intuition that, for a pair of neighbours, the geodesic path between them is close to the euclidean one. Notice that the derivative of the neural network f is Lipschitz continuous,

$$\exists \mathrm{K} \, \forall \mathbf{x}, \mathbf{y}, \; ||\mathrm{J}_\mathbf{x} - \mathrm{J}_\mathbf{y}|| \leq \mathrm{K} \times ||\mathbf{x} - \mathbf{y}||. \quad (9)$$

Equation 9 is equivalent to the Hessian of f being bounded. Under this assumption, if two points $\mathbf{x}$ and $\mathbf{y}$ are close enough, the Jacobian of one point is approximately equal to the other: if $||\mathbf{x} - \mathbf{y}|| \leq \epsilon$, then $\mathrm{J}_\mathbf{x} \approx \mathrm{J}_\mathbf{y}$. As a result, the length between $\mathbf{x}$ and $\mathbf{y}$, for a curve γ, is: $\mathrm{L}(\gamma) \approx \int_\gamma ||\mathrm{J}_\mathbf{x}|| \, d\mathbf{x} \approx ||\mathrm{J}_\mathbf{x}|| \int_\gamma d\mathbf{x}$. Due to the triangular inequality, the shortest path γ^* is then a straight line, and we have: $\mathrm{L}^*(\mathbf{x}, \mathbf{y}) \approx ||\mathrm{J}_\mathbf{x}|| \times ||\mathbf{x} - \mathbf{y}||$.

As a result, under this assumption, if two points are close, the geodesic path can be approximated with a straight line. Note that even though we take the path between two neighbouring points to be a straight line, we do not assume that the Jacobian of the function between the two points is constant.

Choice of the set of points $\mathbf{x}_i$ When using Geodesic IG, the choice of the points $\mathbf{x}_i$ to approximate the geodesic path

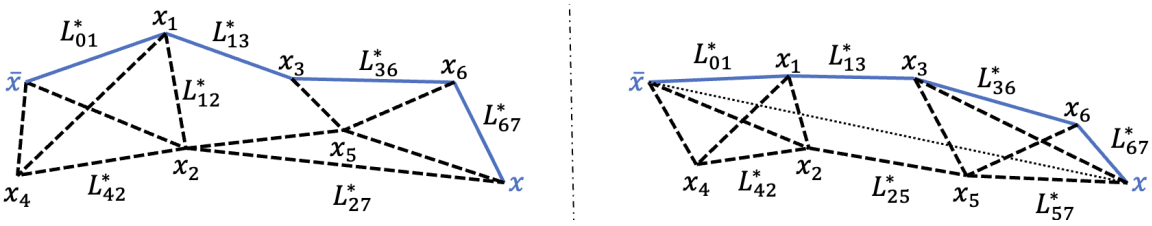

Figure 4. **Method overview.** For an input $\mathbf{x}$, a baseline $\bar{\mathbf{x}}$, and a set of points $\mathbf{x}_i$, we compute the kNN graph using the euclidean distance (dashed lines). For each couple $(\mathbf{x}_i, \mathbf{x}_j)$, we then compute the integrated gradients L^*_{ij} using Equation 6. For clarity, not all L^*_{ij} are present on the figure. 0 and 7 represent $\bar{\mathbf{x}}$ and $\mathbf{x}$ respectively. Using the resulting undirected weighted graph, we use the Dijkstra algorithm to find the shortest path between $\mathbf{x}$ and $\bar{\mathbf{x}}$ (blue continuous lines). On the left, the points $\mathbf{x}_i$ are provided while, on the right, the points are generated along the straight line between $\mathbf{x}$ and $\bar{\mathbf{x}}$ (dotted line).

is of great importance. A wrong choice of points could indeed lead to a bad approximation. We present here two different strategies to select such points.

In a low dimensional setting, such as in the half-moons experiment presented in the introduction, one natural choice is to use points either from the train or from the test set. However, when data is high dimensional, or when the available data is too sparse to be used to approximate the geodesic, others strategies must be used. In these situations, we propose to generate points between an input $\mathbf{x}$ and a reference baseline $\bar{\mathbf{x}}$ using the following equation.

$$\mathbf{x}_i = \bar{\mathbf{x}} + t \times (\mathbf{x} - \bar{\mathbf{x}}) + \mathbf{u}. \tag{10}$$

Here, we uniformly sample points on the straight line between the input and the baseline, using $t \sim \mathcal{U}[0, 1]$, and we add Gaussian noise to these interpolations, using $\mathbf{u} \sim \mathcal{N}(0, \mathrm{I})$, to generate points around this straight line.

The above method of approximating the geodesic using a straight line as a guide is presented as a baseline for practical implementation in high dimensions. However, we believe further research should be undertaken in order to better approximate the geodesic in high dimensional spaces which potentially have a high curvature. For instance, a slightly more advanced method would be to randomly and sparsely seed a larger volume of the space between the input and baseline, then use the shortest path of lowest gradient through these points. After finding such a path, we can use this as a guide, instead of the straight line. Finally, we can repeat this process multiple times and use the path that avoids the most gradients. Such method should yield a better approximation of the geodesic path, albeit requiring more gradients estimations.

Handling disconnected graphs An issue with the graph computed with the kNN algorithm is that it could be disconnected, in which case it could be impossible to compute a path between an input and a baseline. To alleviate this issue, we add so called "bridges" to the graph, as following: for each disconnected component, we add one link between them, specifically between two points of each component having the lowest euclidean distance. An illustration of this method is displayed on Figure 5.

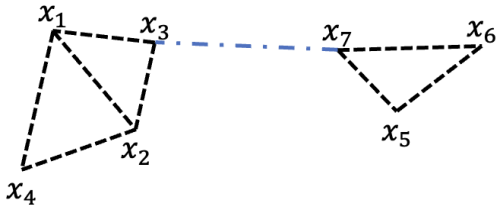

Figure 5. When the kNN graph is disconnected, as illustrated here, it would be impossible to compute Geodesic IG between certain points, for instance $\mathbf{x}_1$ and $\mathbf{x}_5$ here. To solve this, we add a single link between disconnected graphs, here between $\mathbf{x}_3$ and $\mathbf{x}_7$.

However, we stress that this solution is not optimal, and argue that a better way of handling this issue would be to avoid disconnected graphs in the first place. This can be done by increasing the number of neighbours k.

3. Experiments

To validate our method, we performed experiments on two datasets: one is the synthetic half-moons dataset, and the other is the real-world Pascal VOC 2012 dataset.

3.1. Experiments on the half-moons dataset

Here we give the details of the half-moons experiment discussed in the Introduction section. We use the half-moons dataset provided by Scikit learn (Pedregosa et al., 2011) to generate 10,000 points with a Gaussian noise of $\mathcal{N}(0, 0.2)$. The dataset is split into 8,000 training points and 2,000 testing ones. The model used is an MLP.

We measure two indicators of performance for each attribu-

tion method: a lack of artifacts not reflecting the model's behaviour, and a low variation of attributions between close points. To that goal, we use two metrics, **purity** and **standard deviation**, defined in the following way. We know that a well-trained model should classify about half of the data points as upper moon, class 1, and the other half as lower moon, class 0. Such a model should consider both features of each point important for the classification into class 1. Therefore, for such a model, we expect in a good attribution method, the top 50% points as ranked by the quantity $\widetilde{attr}(\mathbf{x}; f) = \sum_{i=0}^{1} |attr_i(\mathbf{x}; f)|$, to be classified as 1 (assuming the baseline is chosen as a point to which the network gives a near-zero score). With this in mind, purity is defined as

$$\text{Purity} = \frac{1}{N/2} \sum_{\mathbf{x},\, \widetilde{attr}(\mathbf{x};f) \in \text{Top } 50\% \text{ of all attr}} \text{argmax}(f(\mathbf{x})), \tag{11}$$

where N is the number of data points. We see that this is the average value of the predicted class labels for half of the points. From the above, we infer that, for a well-trained model, we prefer an attribution method that results in the purity close to 1. In contrast, a random attribution method in this case would result in the purity score of 0.5.

For the second metric, standard deviation, points that belong to the same moon to have similar $\widetilde{attr}(\mathbf{x}; f)$. Therefore, define

$$\text{Std}_i = \text{std}\{\widetilde{attr}(\mathbf{x}; f),\ \text{argmax}(f(\mathbf{x})) \in \text{Moon }_i\}. \tag{12}$$

We expect this standard deviation metric to be low for the points belonging to either class.

In this experiment, we compare the results of attributions from Geodesic IG with the original IG, as well as more recent comparable methods including Enhanced IG (Jha et al., 2020), GradientShap (Lundberg & Lee, 2017), SmoothGrad (Smilkov et al., 2017).

For all of the methods, we use $(-0.5, -0.5)$ as a baseline. Enhanced IG is an improvement over IG where the kNN algorithm is used to avoid computing gradients on paths on out of sample distributions. The chosen number of neighbours for the kNN part of both Enhanced IG and Geodesic IG is 5. We perform an ablation study of this parameter in Appendix B.

The quantitative analysis of the results are given in Table 1, where we see that Geodesic IG significantly outperforms all other methods on all metrics. In particular, notice that all other methods perform relatively poorly on STD_1. This is because for this metric, the integration path has to cross the decision boundary. However, the other methods might cross the decision boundary differently and regardless of the function's gradient, resulting in spuriously different attributions for different points belonging to class 1. In Appendix B we see that the gap between the performance of geodesic IG and the other methods increases as the Gaussian noise of the half-moons increases. To provide better understanding of these results we present more analysis on this dataset in Section 4.

Method	Purity ↑	STD_0 ↓	STD_1 ↓
GradientShap	0.761 (0.158)	1.90 (0.520)	5.01 (0.581)
IG	0.802 (0.142)	1.44 (0.139)	4.915 (0.574)
SmoothGrad	0.800 (0.142)	1.42 (0.139)	4.720 (0.649)
Enhanced IG	0.738 (0.209)	1.23 (0.229)	1.876 (1.10)
Geodesic IG	**0.978** (0.0143)	**0.237** (0.0278)	**0.267** (0.129)

Table 1. Evaluation of different attribution methods on a half-moons dataset with Gaussian noise $\mathcal{N}(0, 0.2)$. The results over 5 different seeds are averaged, with the corresponding standard deviation in brackets. We present in Appendix B more results with different amounts of Gaussian noise.

3.2. Experiments on the Pascal VOC 2012 dataset

Now we would like to test our method on a real-world dataset. To this aim, we use the Pascal VOC 2012 dataset, which consists of labelled images (Everingham et al.). We also use a pre-trained ResNet model to generate predictions to be explained (He et al., 2016). We also perform the analysis on 100 randomly sampled images from the test set.

We compare our method with various baselines: Integrated Gradients, GradientShap, SmoothGrad, InputXGradients (Shrikumar et al., 2016), Lime (Ribeiro et al., 2016), KernelShap (Lundberg & Lee, 2017), Occlusion (Zeiler & Fergus, 2014) Augmented Occlusion (Tonekaboni et al., 2020) and Enhanced IG (Jha et al., 2020).

For both Enhanced IG and Geodesic IG, as the VOC dataset is high-dimensional, we generate points of the graph following the method described in our Method section, using the straight line as a guide. The baseline is chosen here as a uniformly black image. As with the half-moons dataset, we use 5 neighbours for the kNN algorithm.

To evaluate the performance of an attribution method, we use here 3 different metrics:

- **Comprehensiveness** (DeYoung et al., 2019): We mask the top k% most important features in absolute value, and compute the average change of the predicted class probability compared with the original image. A higher score is better as it indicates masking these features results in a large change of predictions.
- **Sufficiency** (DeYoung et al., 2019): We only keep the top k% most important features in absolute value, and compute the average change of the predicted class

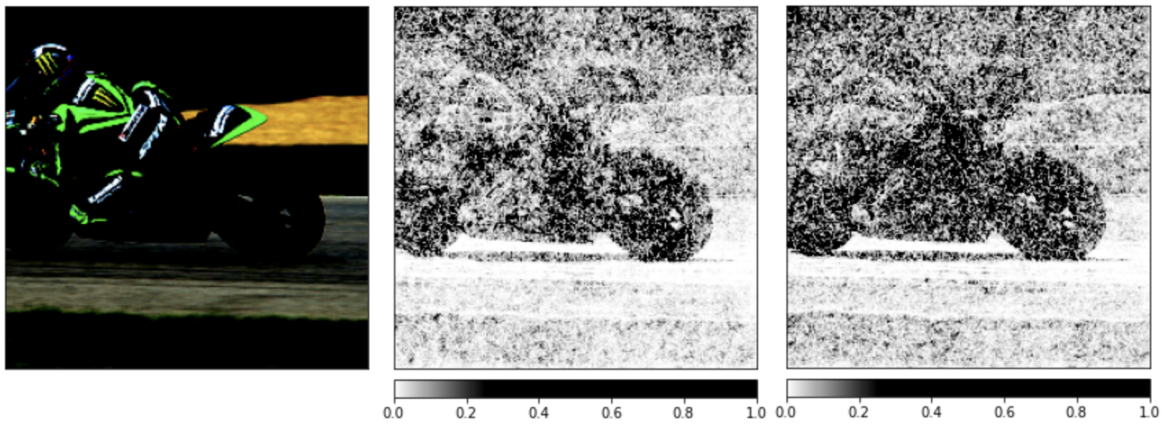

Figure 6. Comparison between Integrated Gradients and Geodesic IG on one image. The predicted class is "motor-scooter". The left figure is the original image, while the other two are heatmaps of respectively the IG (center) and the Geodesic IG (right). We can see that our method seems to reduce the amount of noise and is able to provide a sharper heatmap, compared with the vanilla IG. More comparisons are presented in Appendix C.

Method	Comp ↑	Suff ↓	LO ↓
Input X Gradients	0.360 (0.0280)	0.545 (0.0254)	-2.66 (0.167)
GradientShap	0.425 (0.0130)	0.544 (0.0260)	-3.28 (0.164)
IG	0.427 (0.0072)	0.544 (0.0254)	-3.31 (0.117)
SmoothGrad	0.404 (0.0160)	0.536 (0.0263)	-3.09 (0.222)
Lime	0.197 (0.0139)	0.529 (0.0176)	-1.31 (0.082)
Kernel Shap	0.193 (0.0179)	**0.527** (0.0172)	-1.30 (0.114)
Occlusion	0.340 (0.0263)	0.534 (0.0211)	-2.26 (0.061)
Aug Occlusion	0.352 (0.0177)	0.540 (0.0235)	-2.37 (0.118)
Enhanced IG	0.445 (0.0104)	0.543 (0.0264)	-3.75 (0.125)
Geodesic IG	**0.449** (0.0107)	0.543 (0.0261)	**-3.77** (0.168)

Table 2. Evaluation of different attribution methods on 100 randomly sampled images from the Pascal VOC test set. The metrics are computed by removing or keeping the top 5% most important features. More results are provided in Appendix C.

probability compared with the original image. A lower score is better as it means that these important features are sufficient to retain similar predictions.

- **Log-odds**[2] (Shrikumar et al., 2017): We mask the top k% most important features in absolute value, and measure the negative logarithmic probabilities on the predicted class compared with the original one. Lower scores are better.

Figure 6 provides a qualitative comparison between Geodesic IG and Integrated Gradients, while Table 2 provides a quantitative comparison between Geodesic IG and various other methods. The analysis shows that Geodesic IG is a very powerful method in explaining the model's behavior on the dataset. Table 2 shows that for this dataset, Geodesic IG outperforms almost all other methods in terms of comprehensiveness and log-odds, and is on par with Enhanced IG. However, as we further discuss in section 4, the path of Enhanced IG does not take the model's gradients into account, potentially leading to significant artefacts, and lacks a straightforward way for improvement. On the other hand, Geodesic IG could be further improved by deriving better approximations of the geodesic path, which we discuss in section 2. In this experiment, we have indeed used the simple 'guide' method according to Eq. 10, and we expect that a more sophisticated approximation method would further improve the results.

Furthermore, the Sufficiency results in Table 2 do not seem to discriminate between different explanation methods. However, in Appendix C we present more results, in which Geodesic IG outperforms the original IG on this metric.

[2]This metric should be called *Log-probabilities*. However, since Log-odds is a commonly used name in the literature, we refer to it as Log-odds.

4. Related Work

Approximating geodesic paths is a widely studied area of research, and many methods to do so have been developed. For a comprehensive survey on this subject, please refer to Crane et al. (2020). Our work is specifically inspired from the ISOMAP method (Tenenbaum et al., 2000), a dimensionality reduction method which approximates geodesic paths on a manifold. However, our method differs from ISOMAP in that we weight our graph not using euclidean distance, but the norms of a model's gradients. Our aim is indeed not to model the input space, but to explain a neural network by building paths avoiding high-gradients regions.

As mentioned in Section 3, the idea of using a kNN algorithm to avoid computing gradients on out of distribution data points has also been used in Enhanced Integrated Gradients Jha et al. (2020). However, this method creates a path which is model agnostic, as it does not necessarily avoid high gradients regions. As a result, it can lead to significant artefacts which do not reflect the model's behaviour. To support this argument, we provide an example where this method fails on the half-moons datasets (Figure 7).

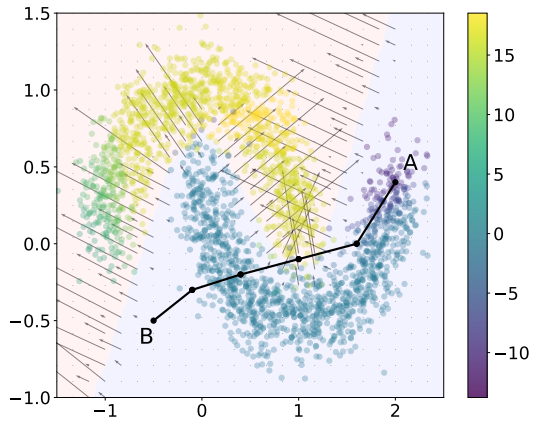

Figure 7. **Enhanced IG attributions.** Enhanced IG computes a kNN algorithm, uses Dijsktra to find the shortest path between an input and a reference baseline, and computes gradients along this path. However, this method is model agnostic and can as a result cross a high gradients region, which is the case in this example, between the input A and the baseline B. Input A therefore has a high attribution which does not reflect the model's true behavior. In this example, the noise is $\mathcal{N}(0, 0.15)$.

The idea of adapting the path according to the model has be proposed by Kapishnikov et al. (2021), calling their method Guided Integrated Gradients. Their method computes this path greedily by selecting around 10% of the features that have the lowest absolute value of the partial derivatives, and switching these features from the baseline's values to the input's values. However, as the authors indicate, such method can create out of distribution data points, with part of their features either matching the input or the baseline. As a result, they add a hyperparameter K, which forces the path to go through K points along the straight line between the input and the baseline. We argue, however, that, by directly approximating the path of least resistance, our method is more principled compared to Guided IG. First, as the latter is a greedy method, it is not possible to compute the gradients in batches, which can make it more computationally expensive. Moreover, it is not clear how to choose the value of K: a too low value could create out of distribution samples, while a too high one would force the path to be close to the straight line and potentially cross high gradients regions. It is similarly not clear why switching 10% of the features, and how to tune this hyperparamer. On the other hand, we argue that Geodesic IG does not have this issue. It has indeed two hyperparamers: the number of points used to approximate the geodesic path, and the number k of nearest neighbors. Yet, the performance of Geodesic IG should improve when both hyperparamers values increase, as, when more points or more neighbors are used, the approximation of the geodesic path should improve. Increasing these values would, however, require more computing and performance needs to be balanced with the amount of compute available.

Dombrowski et al. (2019) rightfully note that a high curvature of the output manifold of the model can lead to vulnerabilities of explanations such as Integrated Gradients, but fail to provide a strong solution which does not undermine the model's performance. Indeed, they theoretically show that replacing ReLU activations with Softplus_β bounds the maximal curvature and reduces the potential to manipulate explanations. However, it is then not clear how to choose the β parameter. A high value would keep the modified model close to the original one, and explanations would still be subjected to manipulations, while, on the other hand, a low value can significantly reduce the accuracy of the model. We discuss this in more details in Appendix A. Moreover, we argue that Geodesic IG should be a strong candidate to tackle potential manipulations, as it uses the geometry of the model.

5. Discussion

We have identified in this paper potential issues with path-based attribution methods: the presence of artefacts and high variation of attributions between close points. To overcome these issues, we have introduced Geodesic Integrated Gradients, an adaptation of the original IG method which integrates gradients not along a straight line, but along the geodesic of a manifold defined by the model.

By avoiding high-gradients regions in the input space, we have shown that Geodesic IG can successfully address these issues. Moreover, it follows all of the axioms defined by Sundararajan et al. (2017). It can also be efficiently estimated by computing gradients in batches.

Moreover, while integrating gradients over an approximated geodesic path should be preferred compared with a straight line, the questions remains in how to generate points to compute this approximation, especially in high-dimensional settings. We have presented one possible method, using the straight line as a guide, but we believe further research should be conducted to improve this approximation while keeping the amount of computing low.

References

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Chen, N., Ferroni, F., Klushyn, A., Paraschos, A., Bayer, J., and Smagt, P. v. d. Fast approximate geodesics for deep generative models. In *International Conference on Artificial Neural Networks*, pp. 554–566. Springer, 2019.

Crane, K., Livesu, M., Puppo, E., and Qin, Y. A survey of algorithms for geodesic paths and distances. *arXiv preprint arXiv:2007.10430*, 2020.

Das, A. and Rad, P. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371*, 2020.

DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., and Wallace, B. C. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*, 2019.

Dombrowski, A.-K., Alber, M., Anders, C., Ackermann, M., Müller, K.-R., and Kessel, P. Explanations can be manipulated and geometry is to blame. *Advances in Neural Information Processing Systems*, 32, 2019.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Holzinger, A., Saranti, A., Molnar, C., Biecek, P., and Samek, W. Explainable ai methods-a brief overview. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pp. 13–38. Springer, 2022.

Jha, A., K Aicher, J., R Gazzara, M., Singh, D., and Barash, Y. Enhanced integrated gradients: improving interpretability of deep learning models using splicing codes as a case study. *Genome biology*, 21(1):1–22, 2020.

Kapishnikov, A., Venugopalan, S., Avci, B., Wedin, B., Terry, M., and Bolukbasi, T. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5050–5058, 2021.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Marcus, G. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ribeiro, M. T., Singh, S., and Guestrin, C. " 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, 2016.

Sap, M., Card, D., Gabriel, S., Choi, Y., and Smith, N. A. The risk of racial bias in hate speech detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1668–1678, 2019.

Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Tenenbaum, J. B., Silva, V. d., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Tonekaboni, S., Joshi, S., Campbell, K., Duvenaud, D. K., and Goldenberg, A. What went wrong and when? instance-wise feature importance for time-series black-box models. *Advances in Neural Information Processing Systems*, 33:799–809, 2020.

Yang, T., Arvanitidis, G., Fu, D., Li, X., and Hauberg, S. Geodesic clustering in deep generative models. *arXiv preprint arXiv:1809.04747*, 2018.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

A. Use of softplus activations

We discuss here the idea of Dombrowski et al. (2019) and study the effect of replacing the ReLU activations by softplus with different values of β on the half-moons experiment. We present our results on Figure 8 and 9. Our analysis shows that, while using softplus instead of ReLU activations can improve (but not necessarily) attribution methods performance, it also reduces the accuracy of the model. Indeed, as β becomes lower, the MLP gets closer to a linear model, and cannot as such properly differentiate the half-moons. As a result, we would recommend using Geodesic IG with the original MLP model instead of using softplus activations. A further analysis would be to compare the robustness of Geodesic IG against adversarial attacks, as performed by Dombrowski et al. (2019).

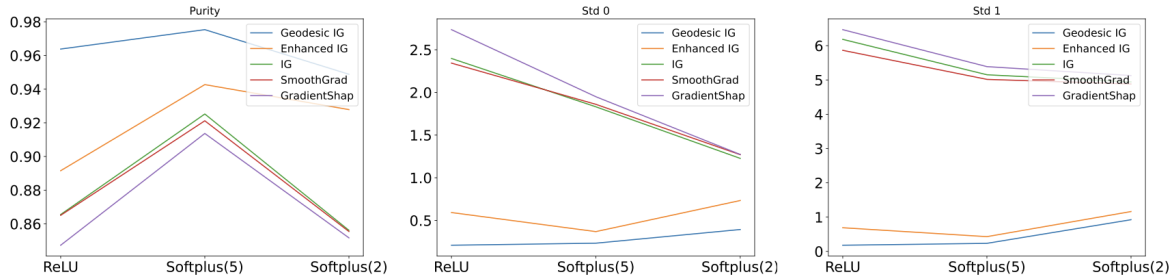

Figure 8. Evaluation of different attribution methods on the half-moons dataset using a MLP with ReLU, Softplus($\beta = 5$) and Softplus($\beta = 2$) activations. Integrated Gradient improves significantly when using Softplus activations, while other method either improve marginally or perform similarly as when using ReLU activations.

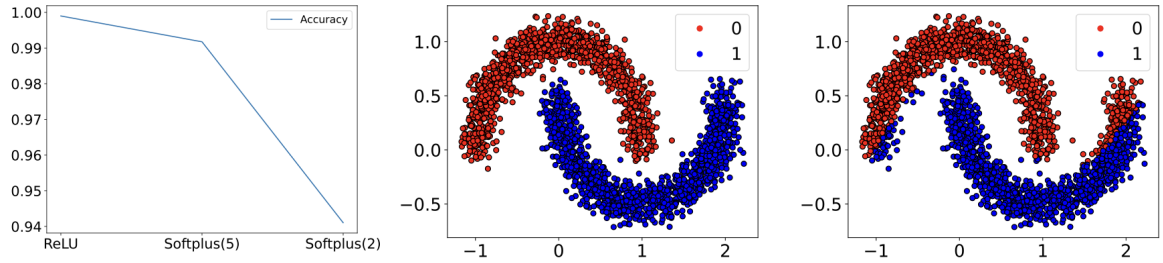

Figure 9. Accuracy of the MLP with ReLU, Softplus($\beta = 5$) and Softplus($\beta = 2$) activations. We see a sharp drop of accuracy when β decreases. We also plot the predictions on the test set of the MLP with ReLU activations (middle plot) and Softplus($\beta = 2$) activations (right plot) to illustrate this drop of accuracy.

B. Additional Half-moons results

We present on Figure 10 more results on the half-moons dataset, using different amounts of noise. We can see that, for certain amount of noise, Enhanced IG dramatically fails, while Geodesic IG performs consistently well on every amount of noise tested here. We believe that the failure of Enhanced IG in the high-noise setting is due to the following reason. As noise increases the points on either moons get closer to each other. As a result, the model loses the property that gradients are only large on the decision boundary and fall rapidly as we move away. Therefore a lot of points are on the high-gradient regions. However, Enhanced IG chooses the path purely based on nearest neighbours, ignoring model gradients. Hence, leading to low purity. This is contrast to geodesic IG, which actively avoids regions of high gradients.

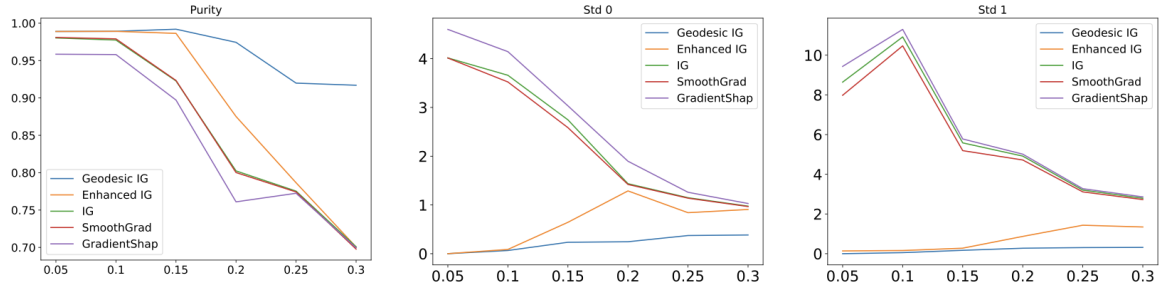

Figure 10. Evaluation of different attribution methods on the half-moons dataset with different amounts of noise: $\mathcal{N}(0, x)$ where x is defined as the axis of each plot.

We also perform here an ablation study using different values of k for the kNN algorithm. We show the results on Figure 11. The results show that increasing k harms the performance of Enhanced IG, leaving the ones of Geodesic IG unchanged. This is probably due to the fact that increasing k allows connections between points further apart, potentially crossing high-gradients regions. While Geodesic IG would not follow such paths, Enhanced IG only uses euclidean distance, and is therefore more likely to generate paths crossing high-gradients regions.

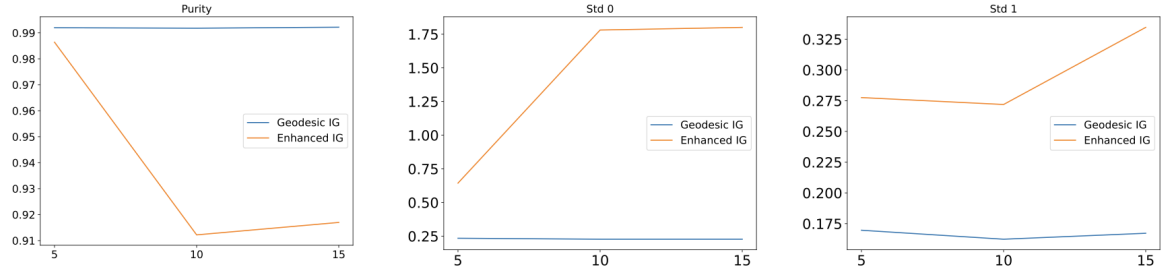

Figure 11. Evaluation of Geodesic IG and Enhanced IG for different values of k in the kNN algorithm. We can see that increasing this parameter harms Enhanced IG performance, while it does not seem to have a major effect on Geodesic IG performance.

C. Additional heatmaps and results on Pascal VOC 2012

We provide here more results using different values of top k% to evaluate Geodesic IG against various other attributions methods. The results are presented on Figure 12. We can see that Geodesic IG and Enhanced IG perform consistently better than other attribution methods across different values of top k%. One exception is SmoothGrad on the Sufficiency metric, which performs better. One improvement of Geodesic IG could be as a result to combine it with the SmoothGrad method, which could potentially yield even better results.

We also qualitatively compare on Figure 13 Geodesic IG with the original IG on 5 different images of the Pascal VOC 2012 dataset. To prevent cherry-piking, these images were randomly sampled. Geodesic IG heatmaps appears to be less blurry than the ones generated with original IG method.

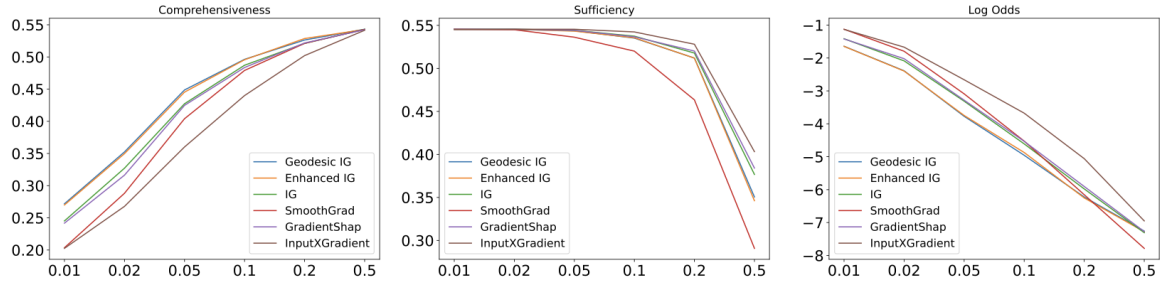

Figure 12. Evaluation of several attribution methods on Pascal VOC 2012 dataset, using different values of top k%.

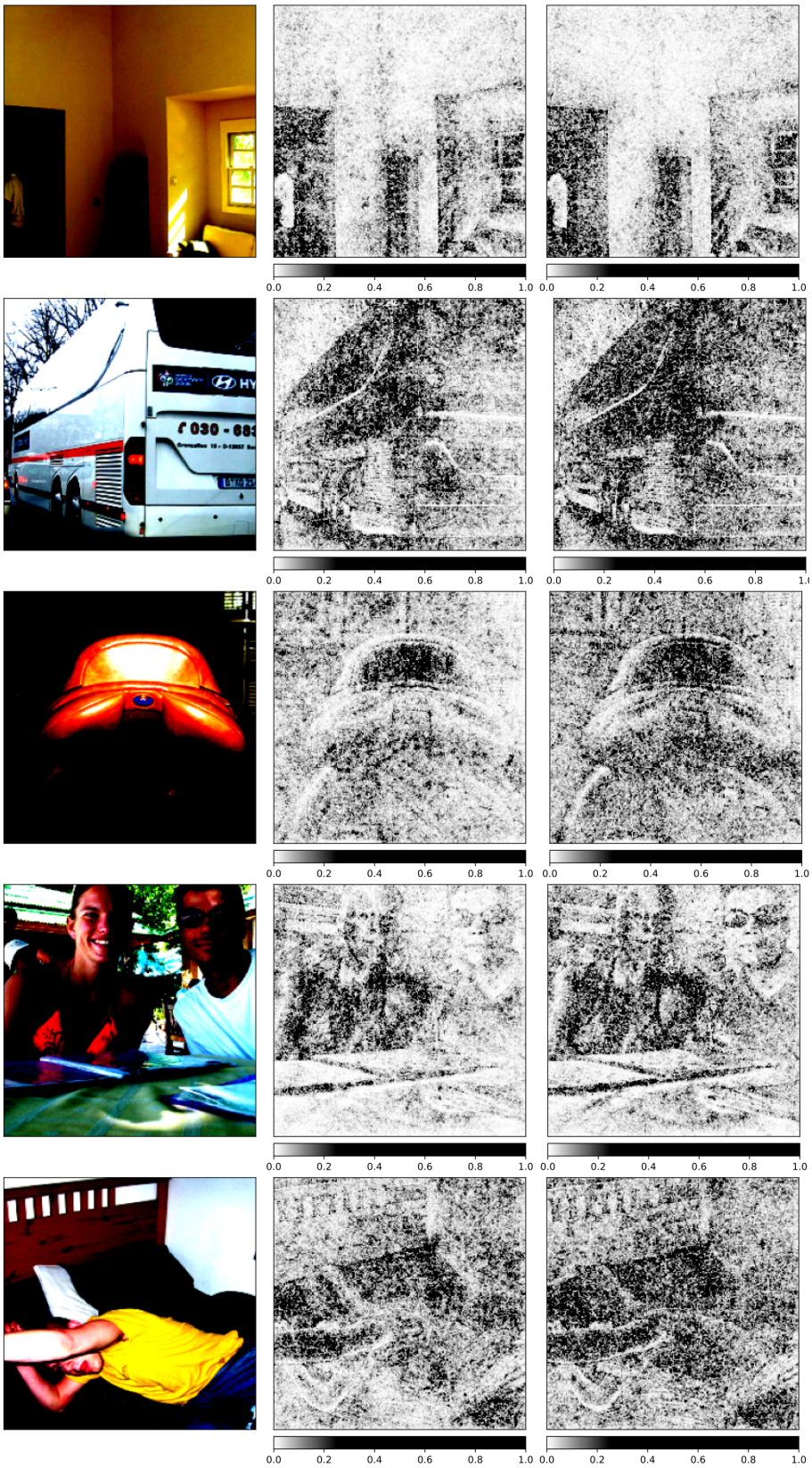

Figure 13. Heatmaps of Integrated Gradients (middle) and Geodesic IG (right) on 5 randomly chosen images from the test set of Pascal VOC 2012. We can see that Geodesic IG heatmaps are sharper compared with the original IG method.