Local Path Integration for Attribution

Peiyu Yang[1], **Naveed Akhtar**[1], **Zeyi Wen**[*2,3], **Ajmal Mian**[1]

[1] The University of Western Australia
[2] Hong Kong University of Science and Technology (Guangzhou)
[3] Hong Kong University of Science and Technology
peiyu.yang@research.uwa.edu.au, naveed.akhtar@uwa.edu.au, wenzeyi@ust.hk, ajmal.mian@uwa.edu.au

Abstract

Path attribution methods are a popular tool to interpret a visual model's prediction on an input. They integrate model gradients for the input features over a path defined between the input and a reference, thereby satisfying certain desirable theoretical properties. However, their reliability hinges on the choice of the reference. Moreover, they do not exhibit weak dependence on the input, which leads to counter-intuitive feature attribution mapping. We show that path-based attribution can account for the weak dependence property by choosing the reference from the local distribution of the input. We devise a method to identify the local input distribution and propose a technique to stochastically integrate the model gradients over the paths defined by the references sampled from that distribution. Our local path integration (LPI) method is found to consistently outperform existing path attribution techniques when evaluated on deep visual models. Contributing to the ongoing search of reliable evaluation metrics for the interpretation methods, we also introduce DiffID metric that uses the relative difference between insertion and deletion games to alleviate the distribution shift problem faced by existing metrics. Our code is available at https://github.com/ypeiyu/LPI.

Introduction

Feature attribution methods are widely used to explain deep visual models (Simonyan, Vedaldi, and Zisserman 2014; Binder et al. 2016; Sundararajan, Taly, and Yan 2017; Smilkov et al. 2017). They compute class-specific importance scores for the input features to provide an attribution map for post-hoc model interpretation. Faithfulness of the computed maps to the model is the corner stone for these methods. Compromising it can lead to misleading model explanations, which can have severe consequences for the high-stake applications, e.g., medical diagnosis, which are often the primary user domains of explainable Artificial Intelligence.

Path-based feature attribution is currently among the most promising attribution techniques (Sundararajan, Taly, and Yan 2017; Erion et al. 2021; Pan, Li, and Zhu 2021). For a given sample, it integrates the model gradients with respect to the input by moving along a certain path in the input space. For instance, the seminal method of Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017) chooses a black image as the reference point, and defines a linear path from that to the input. Integrating the model gradients along that path is used for attribution mapping.

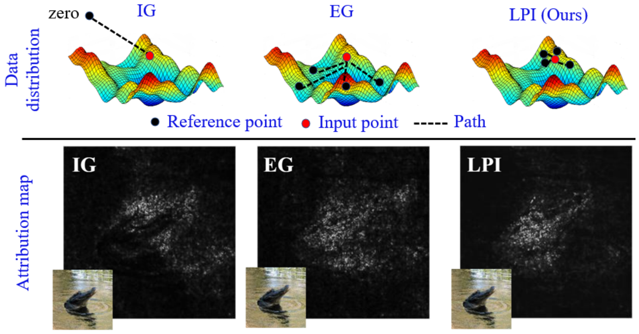

Figure 1: Top: Conceptual difference between the proposed Local Path Integration (LPI) and existing methods. Integrated Gradients (IG) selects a zero image as reference. Expected Gradients (EG) uses random points on the distribution. LPI identifies local distribution of input point and stochastically integrates over the paths defined by local samples. Bottom: IG underestimates importance of darker pixels in the input. EG overestimates the background. LPI consistently computes accurate attribution. Input image is shown for reference.

It is known that path-based mapping satisfies certain theoretical properties (Sundararajan, Taly, and Yan 2017), which makes it particularly attractive for computing model-faithful maps. However, reliability of the path attribution methods heavily depends on the choice of the reference. Similar to the cooperative game theory, the "reference" in these methods represents feature "absence". IG chooses zero input as the reference to simulate the absence. However, the zero reference causes the black pixels of the input to be assigned zero attribution, see Fig. 1. Currently, the debate on the reference selection for the path-based methods is not fully settled (Shah, Jain, and Netrapalli 2021; Pan, Li, and Zhu 2021). Moreover, Srinivas and Fleuret (2019) also showed that path-based methods with the dependence of the input in general do not satisfy a so-called "weak dependence", which can lead them to compute counter-intuitive attribution maps.

Among the closely related methods to our work, Erion et al.

[*] Corresponding author

(2021) proposed an Expected Gradients (EG) method that is able to alleviate the problem related to the choice of the reference. The EG uses multiple references from the model's training data distribution, and integrates over the path defined by them. Though effective, this method often ends up overestimating the importance of irrelevant background in the input - see Fig. 1. There are also other path attribution methods, e.g., (Sturmfels, Lundberg, and Lee 2020; Pan, Li, and Zhu 2021) dedicated to identifying better reference for the path-based attribution. However, optimal choice of the reference still eludes the literature. The problem aggravates when the methods also do not account for the weak dependence property, as noted by (Srinivas and Fleuret 2019).

With the theoretical treatment of the problem, this paper first identifies a link between the weak dependence property and the selection of the reference for path-based attribution. In essence, we find that it is possible to satisfy the weak dependence property when the path is defined by a reference that invokes the same piecewise linear component of the model that is used by the input image. This finding translates to selecting the reference from the same local distribution to which the input sample belongs. Hence, we develop a systematic technique to identify the local distribution for an input using the clusters defined over the model outputs for the training data. To address the ambiguity of feature absence, we draw multiple samples from this distribution and stochastically integrate the gradients along the paths formed by choosing these samples as references.

With extensive experiments on the validation set of ImageNet 2012 (Russakovsky et al. 2015) using two visual classification models, we show that the proposed method of Local Path Integration (LPI) consistently outperforms the existing path-based attribution methods. We also contribute an evaluation metric for reliable performance estimation of the attribution methods. The proposed metric - dubbed DiffID for Difference between Insertion and Deletion games - is comprehensive and alleviates the distribution shift problem in the evaluation process by leveraging the relative difference between the insertion and deletion games (Petsiuk, Das, and Saenko 2018) instead of relying on their absolute values.

The key contributions of this work are as follows.

- It identifies that path-based attribution can satisfy weak dependence (Srinivas and Fleuret 2019) by selecting a reference that invokes the same piecewise linear component of the model as used by the input.
- It devises a process to identify the local distribution for the input.
- It proposes a Local Path Integration scheme that stochastically integrates the model gradients over the paths defined by the references sampled from the local distribution.
- It contributes a comprehensive metric DiffID for reliable performance evaluation of path attribution methods, and establishes the efficacy of our method with the proposed and existing metrics on the ImageNet validation set.

Related Work

In the literature, we find a branch of attribution methods that relies on backpropagating model gradients to quantify the role of input features in model prediction. For instance, Input Gradients (Simonyan, Vedaldi, and Zisserman 2014) is one of the first methods that visualizes feature importance as an attribution map by computing model derivatives with respect to the input. Guided Backpropagation (Springenberg et al. 2015) is a variant of Input Gradients that is obtained by changing the backpropagation rule for the input gradients to generate a cleaner attribution map. Layer-wise Relevance Propagation (LRP) (Bach et al. 2015) assigns importance to the input features based on a backpropagation process that decomposes the prediction layer-by-layer until it reaches the input. Similar to LRP, DeepLIFT (Shrikumar, Greenside, and Kundaje 2017) estimates the attribution scores for all the neurons by comparing the activation at a reference. More recently, Srinivas and Fleuret (2019) decomposed the network into weights and biases to satisfy certain desirable axioms in their FullGrad approach.

Adopting a slightly different strategy from the above, another popular branch of attribution methods estimates feature importance by integrating their attribution score along a path from a reference image to the input. These methods are commonly known as path attribution methods. The first influential approach along this direction is known as Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017), which calculates an attribution map by averaging the gradients along a linear path from a reference to the input. Though axiomatic, IG must attribute zero importance to the input features that accidentally match with the reference image. To solve this blindness problem, Expected Gradients (Erion et al. 2021) integrates the gradients over multiple references, which comes at a considerable computational cost. Addressing the same underlying problem, Adversarial Gradient Integration (Pan, Li, and Zhu 2021) computes the importance score from an adversarial example to the input along the steepest descent path by attacking the target model. Sturmfels, Lundberg, and Lee (2020) compared the impact of different references and suggested further exploration in choosing appropriate reference images for the path attribution methods. Except for finding references, Guided IG (Kapishnikov et al. 2021) chooses features with lowest partial derivatives to form an adaptive integral path for improving the integral path in attribution integration.

Whereas the attribution methods in general are considered effective, recent works (Adebayo et al. 2018; Kindermans et al. 2019; Srinivas and Fleuret 2019) also indicate their fragility. Adebayo et al. (2018) demonstrated that multiple attribution methods fail basic sanity checks. Kindermans et al. (2019) argue that many attribution methods do not ensure an important property – input invariance. Weak dependence (Srinivas and Fleuret 2019) is considered a generalization of input invariance property that can ensure the reliability of attribution methods, which is difficult to guarantee by the current path attribution methods.

Along the attribution techniques, quantification of the estimated importance is also an active topic in the related literature (Samek et al. 2017; Wang et al. 2020; Schulz et al. 2020). Among these metrics, insertion and deletion games (Samek et al. 2017; Ancona et al. 2018; Srinivas and Fleuret 2019) are now widely used to evaluate the methods. They quantify the importance based on image perturbations that insert or remove the most salient pixels in the image to analyze their effect on the model output. Hooker et al. (2019) argue that removing pixels can cause an input shift from the learned distribution, making the reasons behind model performance degradation ambiguous. Hence, they proposed RemOve And Retrain (ROAR) method to measure the output change. Shah, Jain, and Netrapalli (2021) further enhanced ROAR to DiffROAR by measuring the difference of output between the insertion and deletion images. However, their method still requires model retraining, which makes it unclear if the computed scores truly represent the original model. In short, in the contemporary literature, we find a clear room for more reliable and comprehensive evaluation metrics for the attribution methods.

Preliminaries

To help understand our contribution and its significance, we first provide a brief review of the Integrated Gradients (IG) method (Sundararajan, Taly, and Yan 2017) and the notion of weak dependence (Srinivas and Fleuret 2019).

Integrated Gradients

Let $F(.)$ denote the function represented by a neural model. The IG explains its prediction w.r.t. an input x by integrating the gradients along a linear path from a reference x' to x with a step α. Mathematically,

$$\phi_i(x, x') = (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} \mathrm{d}\alpha, \quad (1)$$

where $\phi_i(.,.)$ is the estimated attribution score of the i^{th} feature of x. IG asserts that x' signifies the absence of the corresponding features in x. Hence, it uses a zero image as the reference x'. Due to the axiomatic nature of this assertion - see (Sundararajan, Taly, and Yan 2017), attribution computation along a path from a reference to the model input is a common strategy adopted by numerous attribution methods, e.g., (Erion et al. 2021; Pan, Li, and Zhu 2021; Hesse, Schaub-Meyer, and Roth 2021).

Weak Dependence

Srinivas and Fleuret (2019) identified a desirable property for the attribution methods, known as weak dependence. We can formally define it as below.

Definition 1 (Weak dependence)**.** *Consider a piecewise-linear model $F(.)$ encoding 'n' models defined over the same number of open connected sets $\mathcal{U}_i$ for $i \in [1, n]$.*

$$F(x) = \begin{cases} \boldsymbol{w}_1^T x + b_1, & x \in \mathcal{U}_1 \\ \quad ... & \\ \boldsymbol{w}_n^T x + b_n, & x \in \mathcal{U}_n. \end{cases} \quad (2)$$

For $F(x)$, a function $A(F(x))$ weakly depends on x when this dependence is indirect, via the neighborhood set $\mathcal{U}_i$ of x. That is, $A(.)$ depends on x only through $\boldsymbol{w}_i$, b_i.

In the above, when $A(.)$ implements an attribution method, its weak dependence on the input helps in suppressing counter-intuitive attributions (Srinivas and Fleuret 2019).

Proposed Methodology

Before introducing the proposed methodology of attribution computation, we highlight the limitations of the broader scheme underlying the current path attribution methods.

Limitations of Current Path-Based Attribution

Counter-Intuitive Attribution: The current path attribution scheme faces a pitfall of counter-intuitive attributions (Srinivas and Fleuret 2019). This can be understood by a simple example, where the most fundamental framework in this direction, i.e., IG is applied to a piece-wise linear function.

Example 1 (Counter-intuitive attribution)**.** *Consider the following piecewise-linear function for input x_1, $x_2 \in \mathbb{R}$.*

$$F(x_1, x_2) = \begin{cases} x_1 + 3x_2, & x_1, x_2 \leq 1 \\ 3x_1 + x_2, & x_1, x_2 > 1 \\ 0, & otherwise \end{cases} \quad (3)$$

Take two pairs of points $(x_1, x_2) = (1.5, 1.5)$ and $(x_1, x_2) = (4, 4)$ that satisfy $x_1, x_2 > 1$. In Eq. (3), they are subject to the same linear function $F(x_1, x_2) = 3x_1 + x_2$. Applying IG results in $IG(F(1.5, 1.5)) = (2.5, 3.5)$ and $IG(F(4, 4)) = (10, 6)$, which is misleading because the relative importance of x_1 to x_2 is clearly 3 to 1 for $x_1, x_2 > 1$.

According to Srinivas and Fleuret (2019), this limitation of the path attribution methods stems directly from violating the weak dependence property. We refer to the original work for a detailed discussion on this topic.

Ambiguity of Feature Absence: The axiomatic treatment of path attribution is based on the idea that the reference image x' encodes the absence of the features in x. This helps in better attribution because the gradients for the salient features in x are often already saturated. Relying only on x's gradients can hence misrepresent the feature importance. A path that includes feature absence is able to rectify this problem because moving from absence to presence of an important feature results in sharp model gradients. Integrating over the large gradients naturally results in higher attribution scores for the important features.

Unfortunately, feature *absence* is an ambiguous notion in the context of importance attribution. For instance, absence would normally mean removal of a feature (i.e., pixel in an image). Removing all features leaves a zero image, which is exactly what is used by IG (Sundararajan, Taly, and Yan 2017) as the reference. However, IG visibly underperforms when many of the important pixels in the image are 'black'. Similarly, choosing any other uniform color as the reference will require absence of that color in x - more precisely, in the important pixels of x - for an appropriate attribution. From this perspective, the reference can itself influence the final attribution scores. This also calls into question engineered reference images for the path attribution methods.

Addressing the Limitations

To avoid counter-intuitive attributions, we need a method that accounts for the weak dependence property noted in Def. (1). According to Lemma 1, we can ensure weak dependence of a path attribution method on x if the former is computed for

$\mathcal{U}_i = \Psi(x)$ such that $\mathcal{U}_i$ is the neighborhood of x. This is an important insight that will eventually be exploited by our attribution method discussed in the sections to follow.

Lemma 1: *Following Def. (1), a path attribution method weakly depends on x when x' considers the neighborhood $\Psi(x)$ as the reference, where $\Psi(.)$ is an operator to form an arbitrary neighborhood of x with the same interval endpoints.*

Proof: *For a path attribution method $\bar{\nabla}_{\tilde{x}} \cdot (x - x')$ where $x', \tilde{x} \in \mathcal{U}_i$, we assert $\mathcal{U}_i$ has the same interval endpoints. Since x' and x are in the neighborhood $\mathcal{U}_i$ where 'i' is fixed $\forall x \in \mathcal{U}_i$, the variation $x - x'$ can be omitted. According to Def. (1), input gradients ∇_x satisfy weak dependence. Thereby, weak dependence of path attribution method is ensured.*

Below, we revisit Example 1, while developing a variant of IG in light of Lemma 1.

Example 2 (Intuitive attribution)**.** *Consider again the piecewise linear function defined in Example 1. Given two pairs of points $(x_1, x_2) = (1.5, 1.5)$ and $(x_1, x_2) = (4, 4)$, they satisfy $x_1, x_2 > 1$ and are subject to the linear component $F(., .) = 3x_1 + x_2$. Let $IG^\dagger$ be an IG variant that selects the neighborhood as the reference. Then, the importance for the two pairs of points gets the values $IG^\dagger(1.5, 1.5) = \epsilon(3, 1)$, $IG^\dagger(4, 4) = \epsilon(3, 1)$, where the variable ϵ is a scaling factor.*

The above example shows that it is possible to suppress the counter-intuitive attribution of IG with a suitable selection of the reference. Before proceeding further, we also make a formal remark regarding the issue.

Remark: *Conventional path attribution methods, e.g., IG, violate weak dependence because generally $i \neq j$ for $x \in \mathcal{U}_i$ and $\tilde{x} \in \mathcal{U}_j$, when $\tilde{x} = x' + \alpha(x - x')$, where x' is a zero or arbitrary reference image.*

From the previous section, we know that the second major problem with the path attribution methods is the ambiguity about the *absence* of a feature. It is not our intention to formalize a perfect definition of 'feature absence' in this work. However, we do argue that randomly sampling a number of images from an image distribution defined over the neighborhood of a given input is likely to contain samples that reasonably emulate feature absence w.r.t. the input image. The intuition behind the argument is simple. Assuming feature absence to be a continuous quantifiable notion, a larger absence would require the reference to allow better gradient integration when it is chosen as the starting point of the path. Practically, this demands the pixel values in the reference to be considerably different than the corresponding pixel values in the input. A sufficiently large set of samples in the neighborhood of an image can collectively satisfy this easily. Thus, an attribution can still be well estimated using a large set of references in the neighborhood of the input.

The above discussion suggests that the neighborhood of an input can help in not only mitigating the counter-intuitive results for path attribution but also dealing with saturated features. Keeping in view this intuition, we aim at estimating the attribution as follows

$$\phi_i(x, D_i) = \underset{x' \sim D_i}{\mathbb{E}} (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} \mathrm{d}\alpha \tag{4}$$

where D_i is a distribution defined over the neighborhood of x. Note that, samples from this distribution construct the set $\mathcal{U}_i$ in Lemma 1, where $\Psi(.)$ gets implemented as a distribution sampler for the input x.

Proposed Local Path Integration

The computation in Eq. (4) requires identification of the distribution over the neighborhood of the input. Owing to the large dimensionality of the image space, this identification can be computationally expensive, requiring efficient solution. In the text below, we develop a tractable method for identifying and sampling the desired distribution. Eventually, we compute our attribution scores following Eq. (4). Since we only need to focus on the local neighborhood of the input for this computation, we refer to our method as Local Path Integration (LPI).

Identifying the Neighborhood of the Input: To identify the neighborhood distribution of the input, we first need to identify the overall distribution learned by the model itself. Inspired by the Activation Atlas (Carter et al. 2019), we employ the training samples to achieve that goal. Since the model is learned to fit the training data, the outputs of the training samples can be treated as the points drawn from the distribution learned by the model. At this stage, we use an exhaustive input-output mapping of the model to represent the underlying distribution. We consider logit scores as the outputs due to their larger variability in the coefficient values, which is more desirable for our task. We collect the outputs into a representation set $\mathcal{L}$, and use it as a discretized proxy for the distribution learned by the model. The left part of Fig. 2 illustrates the collection of the outputs.

We need to identify local neighborhoods in the learned distribution. For that, we operate on the output space which comprises manageable feature vectors instead of large images in the input space. Assuming there are n open-connected sets underlying the learned distribution, we employ K-means with Euclidean distance to cluster the outputs in $\mathcal{L}$ into n clusters. We can use the clusters to represent different local distributions $\{D_i | i \in [1, n]\}$ for the neighborhoods. Recall that weak dependence sees the function represented by the model as a piecewise-linear function. Whereas it is relatively easy to approximate an arbitrary non-linear functions as a piecewise linear function, it is hard to identify the point neighborhoods corresponding to the individual piecewise linear components. With output clustering, we can reasonably approximate these neighborhoods because the model behavior remains largely similar for the points falling within a cluster. The middle part of Fig. 2 illustrates the clustering process, where we employ UMAP (Hooker et al. 2019) to project the high-dimensional vectors to 2D space in the figure.

Drawing from the Local Distribution: After identifying different neighborhoods, samples from the same neighborhood can be employed to approach the local distribution learned by the model. Subsequently, we can integrate the gradients over the distribution for an attribution map. However, this computation is intractable. Therefore, we further cluster the outputs in a sub-distribution D_i into m clusters. We collect m samples closest to the cluster centroids to anchor the dis-

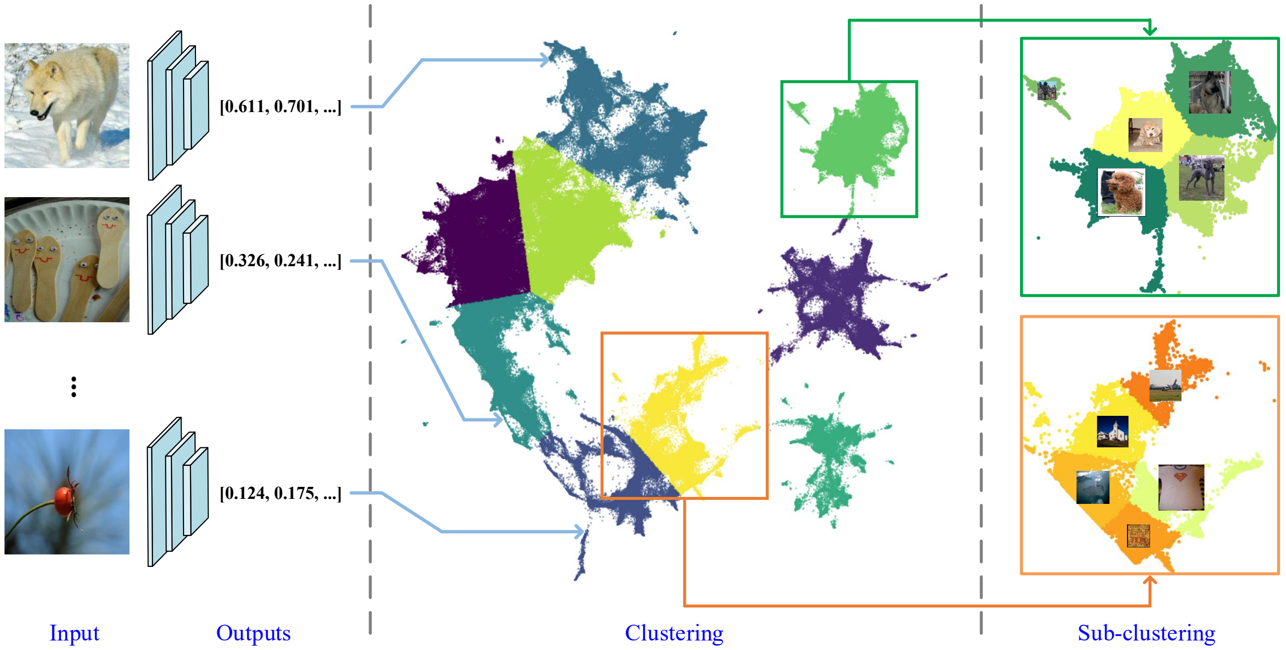

Figure 2: The pipeline of the reference choice. Left: The outputs of the deep model are collected by scanning all the training examples to represent the learned distribution of the target model. Middle: The learned distribution is further divided into different neighborhoods by clustering. Right: Each sub-distribution is further clustered to find representative images for drawing the sub-distribution efficiently. The shown example images are scaled by the corresponding density of the clusters.

tribution D_i. To approximate the probability density function p_{D_i} from the m representative samples, the proportion of the samples in each cluster is used. Using this information, we can efficiently draw from the distribution D_i by sampling from clusters using the corresponding densities. The right side of Fig. 2 illustrates the process of distribution estimation, where the reference image is scaled in proportion to the sample density.

Integration Over Local Paths: We now have all the ingredients for local path integration (LPI) for attribution estimation. For an input image x, the proposed LPI first obtains the output of x. Then, it locates the sub-distribution D_i in which input x resides using the methodology discussed above. Then, it integrates gradients over m references x' of distribution D_i weighted by the density p_{D_i}. This translates to estimating the following.

$$\phi_i(x, D_i) = \int_{x' \sim D_i} \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha p_{D_i}(x') dx', \tag{5}$$

where α is the step over the linear path from the reference to the input. Inspired by Expected Gradients (Erion et al. 2021), we further use the Monte Carlo estimation to speed up the integral calculation by one random step α. We formulate the integral as the Expectation, giving us

$$\phi_i(x, D_i) = \mathop{\mathbb{E}}_{x' \sim D_i, \alpha \sim U(0,1)} (x_i - x'_i) \frac{\partial f(x' + \alpha \times (x - x'))}{\partial x_i} p_{D_i}(x'), \tag{6}$$

where the step α is sampled from a uniform distribution $U(0, 1)$, and the reference x' is sampled from the distribution D_i in which the input x resides. The above attribution provides a tractable approximation of the objective in Eq. (4). It integrates the attribution over local paths identified by the samples of D_i.

Experiments

In this section, we first introduce a new metric to fairly evaluate attribution methods, followed by quantitative results. We note that since our contribution is in path-based attribution, our performance analysis and benchmarking focuses on path attribution methods.

Quantitative Metric – DiffID

A reliable attribution map should assign more importance to the input features that are more relevant to the model prediction. Hence, deleting the most salient pixels of the input image should cause a larger variation in the model output. A converse observation can also be made regarding inserting the more salient features. These observations have inspired the insertion and deletion games (Petsiuk, Das, and Saenko 2018) that measure the model output change by removing $n\%$ of the most or least salient pixels of the input. The insertion and deletion games are widely used to evaluate the performance of the calculated attribution map (Samek et al. 2017; Ancona et al. 2018; Pan, Li, and Zhu 2021).

Though effective, insertion and deletion games have also incurred some criticism in the literature. For instance, Hooker et al. (2019) argued that removing image pixels causes a distribution shift for the input. Therefore, it is unclear whether the performance variation comes from the input shift or removing the informative features. As a remedy, they suggest

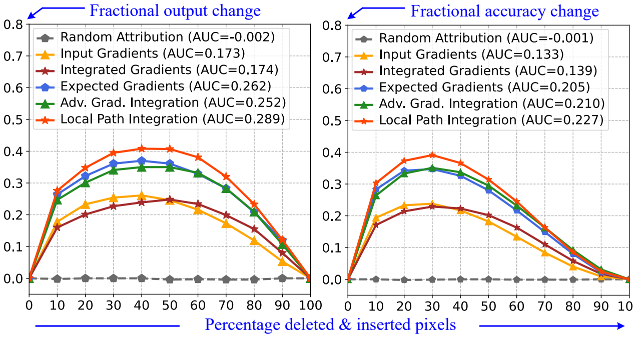

Figure 3: DiffID results on ImageNet validation set for VGG-16. Left: Fractional output change as number of pixels are deleted and inserted for deletion and insertion games. Right: Fractional accuracy change. AUC values are also reported. Higher values are more desirable.

to retrain the model after removing the pixel in their remove and retrain (ROAR) evaluation. However, the target of an attribution method is to faithfully explain a given model, whereas retraining the model changes the model itself. It can be argued that the pursuit of a reliable metric for attribution evaluation is currently still ongoing. We contribute in this direction by proposing to quantify attribution performance with the **Diff**erence between the **I**nsertion and **D**eletion games (DiffID). Whereas this metric is still not completely distribution shift agnostic due to insertion and deletion, it has two-fold benefits. First, it is more comprehensive than the insertion or deletion game alone, which gives a better overall picture. Second, since the shift occurs during both pixel insertion and deletion, focusing on their relative difference instead of their absolute values helps in neutralizing the effects of distribution shift.

To implement DiffID, we also deviate from the deletion game in that instead of replacing a removed pixel with a black pixel, we replace it with the mean-imputation for the input. This is inspired by Sturmfels, Lundberg, and Lee (2020). The intuition is that simply deleting the most salient pixels in the image can cause influential high-frequency edge artifacts. This effect, thereby the resulting distribution shift can be mitigated with the employed strategy. Formally, given an input x and a model $F(.)$, the DiffID evaluates the score of an attribution method as follows.

$$\psi(x, \delta) = F(\mathrm{Ins}(x, 1-\delta)) - F(\mathrm{Del}(x, \delta)), \quad (7)$$

where Ins(.) is the insertion game which inserts $1-\delta$ fraction of the most salient pixels of the input x, and Del(.) is the deletion game which deletes δ fraction of the least salient pixels of x. Note that, the DiffID is zero when $\delta = 0$ and $\delta = 1$, which represent the full image and zero image respectively.

Comparison

We apply DiffID to evaluate the performance of attribution methods on VGG-16 (Simonyan and Zisserman 2015) and ResNet-34 (He et al. 2016) on ImageNet 2012 validation set (Russakovsky et al. 2015). We emphasize that the reported results are averages computed over 50K images, which makes our evaluation extensive. We compare the proposed Local Path Integration (LPI) method with the other popular path attribution methods, including Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017), Expected Gradients (EG) (Erion et al. 2021) and Adversarial Gradient Integration (AGI) (Pan, Li, and Zhu 2021). Due to its popularity, we also include the Input Gradients (Simonyan, Vedaldi, and Zisserman 2014) in our comparison. Moreover, where appropriate, a random attribution is also used as a baseline. For IG, we segment the linear path with 20 steps for the integral. In AGI, the reference is generated by the PGD attack (Madry et al. 2018) with 20 steps and a single random target class. For both LPI and EG, we employ 20 references for each input with one random step. In LPI, we empirically divide the learned distributions into 9 and 7 neighborhoods for VGG-16 and ResNet-34, respectively. Notice that, in the above setup, we account for evaluation fairness by carefully matching the corresponding hyper-parameters of the methods, e.g. both EG and LPI use 20 references and a single step per reference to match the 20 steps in IG and AGI.

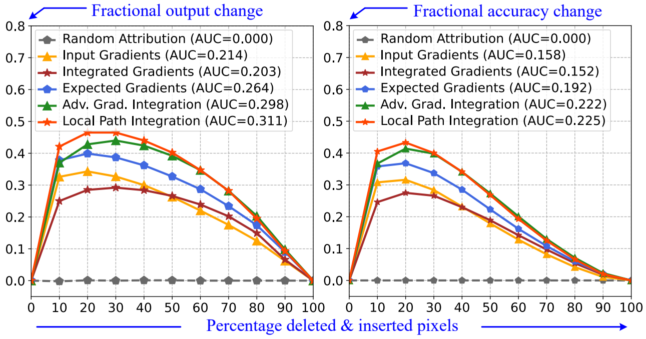

Figure 4: DiffID results on ImageNet validation set for ResNet-34. AUC values are also reported.

Method	Insertion Game		Deletion Game	
	VGG-16	ResNet-34	VGG-16	ResNet-34
Random	0.227	0.355	0.227	0.356
InputGrad	0.418	0.487	0.204	0.175
IG	0.425	0.489	0.222	0.425
EG	0.443	0.526	0.180	0.259
AGI	0.484	0.531	0.184	0.278
LPI	**0.487**	**0.547**	**0.175**	**0.257**

Table 1: The AUC comparison of fractional output changes. In Insertion Game, higher values indicate better performance. In Deletion Game, lower values indicate better performance.

Figure 3, 4(left) show the fractional output change compared to the original output as the δ value for DiffID grows. In addition, Fig. 3,4(right) also show the accuracy change compared to the original input as the δ value grows. The area under curve (AUC) values for LPI and other attribution methods on both VGG-16 and ResNet-34 models are also reported in the legend of the plots. Moreover, in Table **??**, we also report the results for both insertion and deletion games for all methods. All quantitative results show that the proposed

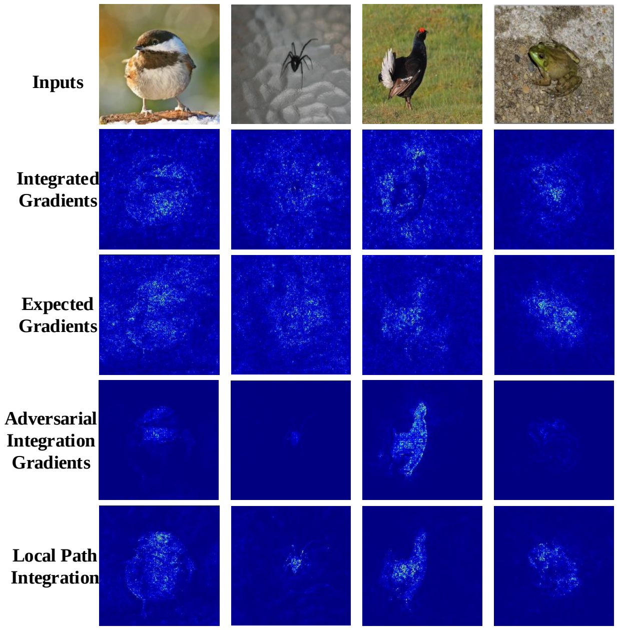

Figure 5: Representative examples for visual comparison of attribution maps computed by the proposed Local Path Integration and the popular attribution methods.

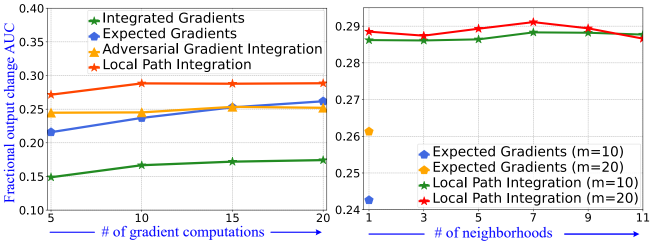

Figure 6: Left: Fractional AUC change w.r.t. the change in number of gradient computations. Right: Fractional AUC change w.r.t. the number of neighborhoods. EG always uses one neighborhood. m is the number of drawn samples. Higher values are more desirable for both plots.

LPI is able to consistently outperform all the related methods. The improved performance is attributed to the careful design of the approach that is directed to address the central weaknesses of the path-based attribution paradigm.

Figure 5 provides representative visualizations of the saliency maps produced by the proposed LPI and the other attribution methods used in our experiments for ResNet-34. We can observe that the attribution map of LPI aligns well with the foreground object. It is also observable that attributions computed by the other methods can be counter-intuitive. For example, the saliency map generated by the Integrated Gradients is blinding the black pixels of the foreground objects. Similarly, the Expected Gradient maps are often overestimating the importance of the background.

Discussion

Attribution estimation of path-based methods, including ours, relies on performing a number of gradient computations. As the key hyper-parameter, it calls for analysing the effects of changing its value on the performance of the methods. We provide this analysis here. In Fig. 6(left), we plot the AUC for the path attribution methods against the number of gradient computations used by the methods. The plots are provided for ResNet-34, using ImageNet validation set. From the figure, it can be seen that the proposed LPI consistently maintains a performance advantage against all methods with respect to the number of gradient computations. Interestingly, Expected Gradients (EG) is able to show some improvements in performance when more computations are allowed. The reason is, each new computation for EG gets performed using a new sample from the training set of the model. With more sampling, it is likely to also pick samples in the neighborhood of the input, which can improve its scores. Since our LPI already starts with the local neighborhood, it is already able to achieve better AUC value than EG with 20 samples, using just 5 samples.

We note that our LPI takes advantage from pre-computing local distributions for the model. However, these one time computations are offline. On the other hand, identifying the sub-cluster of an input at the test time adds negligible amount of computations as compared to attribution estimation. Another interesting hyper-parameter involved in LPI is the number of the neighborhoods used to compute the attribution map. Whereas our LPI is able to explicitly define this, we can also relate this hyper-parameter to EG, where the neighborhood size always remains one, but the same number of images may get picked up from arbitrary local neighborhoods. In Fig. 6(right), we also plot the change in AUC for LPI by varying the number of neighborhoods. The results are also for ResNet-34 on ImageNet validation set. A certain level of consistency is visible in the LPI results with respect to the parameter value choice. Also, the performance of LPI is much better than EG when we use just one image in the neighborhood.

Conclusion

For path attribution methods, violation of weak dependence and ambiguous reference image selection causes counter-intuitive attributions and compromised performance. Addressing this, we introduced a Local Path Integration (LPI) method that focuses on the local neighborhood of an input to select the reference used to integrate the model gradients. We identified the local neighborhood with an offline mapping of the distribution learned by the model. Indexing into that distribution with clustering, we sample a set of images from the local distribution and stochastically integrate the gradients over them to compute the attribution scores. Our strategy is motivated by theoretical insights to address the violation of weak dependence property by the method. We also contributed a comprehensive metric for attribution performance evaluation. Our extensive results on ImageNet validation set and comparison with other path attribution methods establish the efficacy of our technique.

Acknowledgments

This research was supported by ARC Discovery Grant 190102443. Professor Ajmal Mian is the recipient of an Australian Research Council Future Fellowship Award (project number FT210100268) funded by the Australian Government. Dr. Naveed Akhtar is the recipient of Office of National Intelligence, National Intelligence Postdoctoral Grant (project number NIPG-2021-001) funded by the Australian Government.

References

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I. J.; Hardt, M.; and Kim, B. 2018. Sanity Checks for Saliency Maps. In *Advances in NeuralInformation Processing Systems, NeurIPS*.

Ancona, M.; Ceolini, E.; Öztireli, C.; and Gross, M. 2018. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *International Conference on Learning Representations, ICLR*.

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

Binder, A.; Montavon, G.; Lapuschkin, S.; Müller, K.-R.; and Samek, W. 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks, ICANN*.

Carter, S.; Armstrong, Z.; Schubert, L.; Johnson, I.; and Olah, C. 2019. Activation atlas. *Distill*.

Erion, G.; Janizek, J. D.; Sturmfels, P.; Lundberg, S. M.; and Lee, S.-I. 2021. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

Hesse, R.; Schaub-Meyer, S.; and Roth, S. 2021. Fast Axiomatic Attribution for Neural Networks. *Advances in NeuralInformation Processing Systems, NeurIPS*.

Hooker, S.; Erhan, D.; Kindermans, P.; and Kim, B. 2019. A Benchmark for Interpretability Methods in Deep Neural Networks. In *Advances in NeuralInformation Processing Systems, NeurIPS*.

Kapishnikov, A.; Venugopalan, S.; Avci, B.; Wedin, B.; Terry, M.; and Bolukbasi, T. 2021. Guided integrated gradients: An adaptive path method for removing noise. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

Kindermans, P.-J.; Hooker, S.; Adebayo, J.; Alber, M.; Schütt, K. T.; Dähne, S.; Erhan, D.; and Kim, B. 2019. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 267–280. Springer.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations, ICLR*.

Pan, D.; Li, X.; and Zhu, D. 2021. Explaining deep neural network models with adversarial gradient integration. In *International Joint Conference on Artificial Intelligence, IJCAI*.

Petsiuk, V.; Das, A.; and Saenko, K. 2018. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *British Machine Vision Conference, BMVC*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.

Samek, W.; Binder, A.; Montavon, G.; Lapuschkin, S.; and Müller, K. 2017. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Trans. Neural Networks Learn. Syst.*, 28(11): 2660–2673.

Schulz, K.; Sixt, L.; Tombari, F.; and Landgraf, T. 2020. Restricting the Flow: Information Bottlenecks for Attribution. In *International Conference on Learning Representations, ICLR*.

Shah, H.; Jain, P.; and Netrapalli, P. 2021. Do Input Gradients Highlight Discriminative Features? *Advances in NeuralInformation Processing Systems, NeurIPS*.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning, ICML*.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Workshop on International Conference on Learning Representations, ICLR*.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations, ICLR*.

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

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. A. 2015. Striving for Simplicity: The All Convolutional Net. In *International Conference on Learning Representations, ICLR*.

Srinivas, S.; and Fleuret, F. 2019. Full-Gradient Representation for Neural Network Visualization. In *Advances in NeuralInformation Processing Systems, NeurIPS*.

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

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning, ICML*.

Wang, Z.; Mardziel, P.; Datta, A.; and Fredrikson, M. 2020. Interpreting interpretations: Organizing attribution methods by criteria. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW*.