MFABA: A More Faithful and Accelerated Boundary-Based Attribution Method for Deep Neural Networks

Zhiyu Zhu[1], Huaming Chen[1*], Jiayu Zhang[2], Xinyi Wang[3], Zhibo Jin[1], Minhui Xue[4]
Dongxiao Zhu[5], Kim-Kwang Raymond Choo[6]

[1]The University of Sydney
[2]SuZhouYierqi
[3]University of Malaya
[4]CSIRO's Data61
[5]Wayne State University
[6]University of Texas at San Antonio

Abstract

To better understand the output of deep neural networks (DNN), attribution based methods have been an important approach for model interpretability, which assign a score for each input dimension to indicate its importance towards the model outcome. Notably, the attribution methods use the axioms of sensitivity and implementation invariance to ensure the validity and reliability of attribution results. Yet, the existing attribution methods present challenges for effective interpretation and efficient computation. In this work, we introduce MFABA, an attribution algorithm that adheres to axioms, as a novel method for interpreting DNN. Additionally, we provide the theoretical proof and in-depth analysis for MFABA algorithm, and conduct a large scale experiment. The results demonstrate its superiority by achieving over 101.5142 times faster speed than the state-of-the-art attribution algorithms. The effectiveness of MFABA is thoroughly evaluated through the statistical analysis in comparison to other methods, and the full implementation package is open-source at: https://github.com/LMBTough/MFABA.

Introduction

Deep learning (DL) has shown prominent performance in various areas of computing tasks, such as image classification (Li 2022), semantic segmentation (Mo et al. 2022), object detection (Zaidi et al. 2022), and text classification (Minaee et al. 2021). A wide range of applications in practice have demonstrated its effectiveness, unfortunately without much contextual explanation of the decision process. This has led to a severe crisis towards the trustworthiness of DL models given the facts of poor interpretability of results, intractability of model errors, and the difficulty in tracing model behaviours (Janik, Sankaran, and Ortiz 2019). It remains challenging for researchers to obtain a better understanding of complicated models, especially those based on multiple layers and designed for nonlinear learning.

Recently, the attribution methods have been proposed as one of the most promising means to solve this problem, which can find the causal relationship between the inputs and outputs. In general, there are two fundamental axioms proposed in Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017): Sensitivity, and Implementation Invariance. Sensitivity requires good capability of differing the feature and prediction for every input, which ensures that the input information can be correctly attributed for the predictions. In addition, a method that satisfies Implementation Invariance defines that two neural networks with the same input and output values are functionally equivalent regardless the implementation details. The attribution method should retain the same results given two identical networks.

Different from IG using integration to calculate the contribution, Expected Gradient (EG) method (Erion et al. 2021) introduces prior knowledge and uses it as a prior probability distribution for feature attribution, proposing expectation gradients to calculate the importance of an input feature to the output. Boundary-based Integrated Gradient (BIG) method is one of the first methods that uses adversarial attacks with a linear attribution path to identify appropriate decision boundaries for interpretation (Wang, Fredrikson, and Datta 2021). However, the overall performance is limited due to the linear attribution path of IG algorithm (Jin et al. 2023), and BIG requires much more time for computation.

Adversarial Gradient Integration (AGI) method (Pan, Li, and Zhu 2021) exploits the gradient information of adversarial examples to compute the contribution of all input features by integrating the gradient along the non-linear path with the steepest ascent. While AGI does not depend on the choice of reference points in IG, it employs targeted adversarial attacks to discern optimal decision boundaries. Consequently, AGI incurs significant computational costs, particularly in complex tasks and models where extensive gradient integrations are required. The interpretation performance is further hindered if noise or outliers exist in the input data.

To address the noise pixels generated by IG in regions where the prediction category is irrelevant, the Guided Integrated Gradients (GIG) method (Kapishnikov et al. 2021) sets all contributions in the region specified in the feedforward process to zero by guided gradients and only the target region of the network output needs to be considered. While

*Corresponding author: huaming.chen@sydney.edu.au

only effective for limited datasets, GIG has a high demand on computational resources and time cost.

Thus, current attribution algorithms is yet to provide accurate attribution results and require significant computational cost. To tackle these issues, in this work, we propose a new attribution method, called More Faithful and Accelerated Boundary-based Attribution method (MFABA) which attempts to exploits part of the adversarial attack nature for a faster and more effective attribution. MFABA proposes a novel idea based on the second-order Taylor expansion of the loss function in addition to IG, which attribution results demonstrate a more faithful performance. We also investigate the attack from the attribution method to explore the limitations for the linear paths.

In summary, the contributions of this paper are as follows:

- A novel attribution method MFABA is proposed which demonstrate a state-of-the-art performance;
- A detailed derivation of attribution validity and an axiomatic are presented for MFABA;
- The definition of attack in MFABA is provided, and we substantiate the superiority of MFABA through evidence;
- The replication package of MFABA is released.

Background

From Attribution to Interpretability

The interpretability of DL model refers to the ability of explain the predictions and decisions made by the model in a way that human can understand (Zhang and Zhu 2018). For the attribution method in DNN, an exact one-to-one correspondence between the input and output should be provided (Ancona et al. 2017). Thus, attribution method is considered an exclusive set of methods providing the interpretability for DL models. Other general interpretable algorithms may not meet the axiomatic requirements of Sensitivity and Implementation Invariance (Sundararajan, Taly, and Yan 2017).

For most visual related tasks, attribution methods target on the corresponding information for the feature and prediction. It has thus been challenging for comprehensive evaluation with human intuition, which is more applicable with other interpretation approaches, such as Grad-CAM (Selvaraju et al. 2017) and Score-CAM (Wang et al. 2020). These methods are gradient-based methods, among which Grad-CAM faces different challenges like gradient saturation (Ramaswamy et al. 2020), gradient disappearance (Zhang, Wang, and Tang 2019), and low performance for both the coarse-grained heat map generated at the deep level and the fine-grained heat map generated at the superficial level (Choi, Choi, and Rhee 2020). Similarly, Score-CAM aims to bypass the gradients reliance and can obtain the weights of each activation map through forward propagation of scores on target classes. Eventually a linear combination of the weights and activation map will be obtained. However, both methods fail to provide an accurate attribution for the features. Only the intermediate layers of the networks are interpreted, yielding intuitively interpretable results at the corresponding layer. Yet, the methods do not satisfy the two fundamental axioms of the attribution methods.

Saliency Map (SM) (Simonyan, Vedaldi, and Zisserman 2013; Patra and Noble 2020) is one earliest attribution method aiming at the visualisation of the particular features related to model outputs. In details, the partial derivative of the loss function $\partial f(x)$ is leveraged to calculate the degree of importance of x. SM suffers from gradient saturation and does not satisfy the axiom of Sensitivity. For example, in a simple neural network $f(x) = 1 - \mathrm{ReLU}(1-x) = \begin{cases} x, x < 1 \\ 1, x \geq 1 \end{cases}$, the attribution result could be 0 at $x = 0$ or $x = 2$. However, it will be none for $\partial f(x)$. To address this issue, Integrated Gradient (IG) (Sundararajan, Taly, and Yan 2017) integrates the gradients over different paths to obtain the degree of contribution of the non-zero gradient in the non-saturated region. However, it suffers from: (1) poor results due to the choice of baseline may lead to significant deviations (Pan, Li, and Zhu 2021). It is not feasible to find an appropriate baseline for various tasks. (2) expensive computational process, which requires multiple rounds of propagation to obtain approximate integration results. (3) ineffective representation of sample transformation path from the selected gradient path, presenting as an attack conflict issue.

Other methods, like FullGrad method (Srinivas and Fleuret 2019), aim to use local gradient information to interpret DNN internal structure. Distilled Gradient Aggregation (DGA) method (Jeon, Jeong, and Choi 2022) considers the linear regions of decision boundaries based on intermediate local attribution for a sequence of meaningful baseline points. LIME algorithm (Ribeiro, Singh, and Guestrin 2016) amalgamates approximation techniques with weighted sampling methods to construct a local model to generate interpretable predictions from the model classifier. Shapley Additive Explanations (SHAP) algorithm (Lundberg and Lee 2017) computes feature contribution to the prediction outcome using Shapley values and subsequently ranks their importance, thereby achieving both local and global interpretation of the model. However, these methods still face the problem of low accuracy and slow attribution speed. Furthermore, LIME and SHAP tend to emphasize axioms that ensure 'local faithfulness' or 'local accuracy' of explanations, deviating from strict and complete attributions based on sensitivity and implementation invariance axioms.

Gradient-Based Adversarial Attack

Adversarial Attack Algorithms Adversarial attack aims to find the minimum perturbation of input to deviate the output result. In our work, the baseline for finding the attribution and the computational process for optimizing the attribution algorithm are the same. We firstly discuss their relationships and the differences.

$$x_j = x_{j-1} + \eta \operatorname{sign}\left(\nabla_x L(\theta, x, y)\right) \tag{1}$$

$$x^{t+1} = \Pi_{x+\mathcal{S}}\left(x^t + \alpha \operatorname{sign}\left(\nabla_x L(\theta, x, y)\right)\right) \tag{2}$$

Eq. 1 represents the attack process for FGSM (Goodfellow, Shlens, and Szegedy 2014), where a clamp gradient ascent method is used to perform the attack. For I-FGSM (Kurakin, Goodfellow, and Bengio 2018), multi rounds of FGSM attacks are conducted to identify the optimal perturbation direction. Eq. 2 represents the attack process for PGD, where

the distance of the attack is limited by mapping the attacks to the circular space of the data, thereby making the attack result as close as possible to the actual result.

Definition of Successful Attack In the classification task, x_0 belonging to the target category of C and an adversarial sample x_n producing an output category different from C denotes a successful attack, subject to L_P norm for x_0 and x_n is less than ε. Attack samples with larger deviations will be considered as failed attack, as shown in Eq. 3.

$$m(x_0) = C \text{ and } m(x_n) \neq C \text{ and } \|x_0 - x_n\|_p < \varepsilon \quad (3)$$

Method

In this section, we first present the derivation of gradient ascent, with the axiomatic proof for our attribution method. Secondly, we discuss the different methods of 'sharp' and 'smooth' gradient ascent for MFABA.

Theoretical Derivation

Gradient Ascent Method Inspired from BIG and AGI, we consider that gradient ascent of the loss function can push the adversarial samples across the decision boundary of the model. The model's inference response on these adversarial samples is the key information for interpretation. Following Eq. 4-5 present the gradient ascent process using first-order Taylor expansion. L is the loss function.

$$L(x_j + \alpha d) = L(x_j) + \alpha g_j^T d + \varepsilon \quad (4)$$

$$L(x_j + \alpha d) > L(x_j) \quad s.t. \quad g_j^T \cdot d > 0 \quad (5)$$

In which, $g_j = \frac{\partial L(x_j)}{\partial x_j}$, d is an update direction vector with the same dimension as x_j. We can get $L(x_j + \alpha d) > L(x_j)$ if $g_j^T \cdot d > 0$, where $\cdot$ represents dot product. Then we use $x_{j+1} = x_j + \alpha d$ or $x_{j+1} = x_j + \alpha sign(d)$ to update the adversarial sample. The $sign$ function here meets the decoupling requirements of adversarial attacks. To achieve this, the scalar learning rate α for the gradient ascent process will be minimum.

Since the first-order Taylor expansion only takes into account the gradient (first derivative) but ignores curvature (second derivative) of the loss function, it offers a less comprehensive source of information compared to the second-order Taylor expansion.To more accurately depict the local behavior of the model in the vicinity of a given input point, particularly accounting for the non-linear impact of features, we consider that the corresponding function L can be transformed with the second-order Taylor expansion at the point x_j in our MFABA. Next is the derivation of how to obtain the attribution of each model input in MFABA.

MFABA Mathematical Derivation Here, we list second-order Taylor expansion of Eq. 4 during gradient ascent:

$$L(x_j) = L(x_{j-1}) + \frac{\partial L(x_{j-1})}{\partial x_{j-1}}(x_j - x_{j-1}) + \frac{1}{2}\frac{\partial^2 L(x_{j-1})}{\partial x_{j-1}^2}(x_j - x_{j-1})^2 + \varepsilon \quad (6)$$

Eq. 6 indicates that second-order Taylor expansion can be performed when x_j and x_{j+1} are close.

$$\sum_{j=1}^{n} L(x_j) = \sum_{j=0}^{n-1} L(x_j) + \sum_{j=0}^{n-1} \frac{\partial L(x_j)}{\partial x_j}(x_{j+1} - x_j) + \sum_{j=0}^{n-1} \frac{1}{2}\frac{\partial^2 L(x_j)}{\partial x_j^2}(x_{j+1} - x_j)^2 \quad (7)$$

$$L(x_n) - L(x_0) = \sum_{j=0}^{n-1}\left(\frac{\partial L(x_j)}{\partial x_j}(x_{j+1} - x_j) + \frac{1}{2}\frac{\partial^2 L(x_j)}{\partial x_j^2}(x_{j+1} - x_j)^2\right) \quad (8)$$

In Eq. 7 and Eq. 8, we derive the approximate derivation relationship from Eq. 6, and ε is omitted.

$$\frac{\partial^2 L(x_j)}{\partial x_j^2}(x_{j+1} - x_j)^2 = (\frac{\partial L(x_{j+1})}{\partial x_{j+1}} - \frac{\partial L(x_j)}{\partial x_j})(x_{j+1} - x_j) = \triangle x^T H \triangle x = \triangle x^T \cdot \begin{bmatrix} h_{11} \cdot \triangle x^1 + h_{12} \cdot \triangle x^2 + ... + h_{1n} \cdot \triangle x^n \\ \cdots \\ h_{n1} \cdot \triangle x^1 + h_{n2} \cdot \triangle x^2 + ... + h_{nn} \cdot \triangle x^n \end{bmatrix} \quad (9)$$

In Eq. 9, we use the Hessian matrix H to calculate the second-order derivative part in the Taylor expansion.

We replace the second derivative in Eq. 8 with the Hessian matrix proposed in Eq. 9, and finally Eq. 10 is as follows:

$$\begin{aligned} L(x_n) - L(x_0) &= \sum_{j=0}^{n-1}\left(\frac{\partial L(x_j)}{\partial x_j}(x_{j+1} - x_j) + \frac{1}{2}\left(\frac{\partial L(x_{j+1})}{\partial x_{j+1}} - \frac{\partial L(x_j)}{\partial x_j}\right)(x_{j+1} - x_j)\right) \\ &= \sum_{i=0}^{p}\left(\sum_{j=0}^{n-1} \frac{\frac{\partial L(x_j)}{\partial x_j^i} + \frac{\partial L(x_{j+1})}{\partial x_{j+1}^i}}{2}(x_{j+1}^i - x_j^i)\right) \end{aligned} \quad (10)$$

where p indicates the size of the input dimension. And x_j denotes the sample at the j-th gradient ascent, x_0 and x_n are the original and adversarial sample, respectively. Eq. 10 indicates that the difference between $L(x_0)$ and $L(x_n)$ can be seen as the sum of the attribution at each position. In other words, whenever the output values change between them, attribution of non-zero outcomes will be calculated, which meets the axiom of Sensitivity. By iteratively performing gradient ascent and computing the sum of attributions at each position, we can observe how these perturbed features influence the decision-making behavior of the model. Thus, for $\sum_{j=0}^{n-1} \frac{\frac{\partial L(x_j)}{\partial x_j^i} + \frac{\partial L(x_{j+1})}{\partial x_{j+1}^i}}{2}(x_{j+1}^i - x_j^i)$, it can be seen as an attribution on the i-dimensional input.

Axiomatic Proof

Following we discuss the axioms satisfied in MFABA.

Definition of Sensitivity An attribution method satisfies Sensitivity(a) if for every input and baseline that differ in one feature but have different predictions then the differing feature should be given a non-zero attribution.

According to Eq. 8, the sum of all imputations is $L(x_n) - L(x_0)$. Non-zero imputation results are always calculated when x_0 and x_n lead to a change in L. Therefore our method follows the axiom.

Definition of Implementation Invariance A method that satisfies Implementation Invariance should ensure that two neural network attributions with the same input and output values are consistent. It is clear that the computational processes in MFABA follow the chain rule of gradients, which meets the definition of implementation invariance (Sundararajan, Taly, and Yan 2017).

Attribution Method in MFABA

Based on Eq. 10, the attribution corresponding to the i-dimensional input can be expressed as follows:

$$MFABA(x^i) = \sum_{j=0}^{n-1} \frac{1}{2} \left(\frac{\partial L(x_j)}{\partial x_j^i} + \frac{\partial L(x_{j+1})}{\partial x_{j+1}^i} \right) \cdot (x_{j+1}^i - x_j^i) \tag{11}$$

In order to achieve best attribution results, the Taylor expansion needs to ensure that x_{j+1}^i is close to x_j^i. We further use the approximation $\left(\frac{\partial L(x_{j+1})}{\partial x_{j+1}^i} - \frac{\partial L(x_j)}{\partial x_j^i} \right)$ to replace the Hessian matrix, as it may be computationally expensive. Thus, little additional computational time is needed since there is no additional forward and backward propagation.

The adversarial sampling will stop when the decision boundary is found (e.g., a category shift in a classification problem) to avoid potential bias in a sample. We have pratically set a maximum n for gradient ascending step to mitigate additional computing costs in the absence of identified decision boundaries. Meanwhile, the function L is broadly explored in experiments with comparison with BIG and IG, in which neural network output value is selected for attribution. However, we observed that a negative attribution may be generated. Suppose we have a toy sample for a three-classification task, the adversarial attack attempts to alter the output value of the model from [0.4, 0.5, 0.55] to [0.5, 0.65, 0.6]. Here 0.4, 0.5 and 0.55 represent the output values of class A, B and C respectively. It is obvious that after perturbation, the final output class of the model changes from C to B. But the confidence value for class C is actually increased, resulting in attribution errors when other models like IG and BIG use the probability results obtained before softmax function as the model output. We observe that the probability results obtained via softmax function can highlight the correct reduced probabilities for classification, which helps to mitigate the attribution issue. Thus, in MFABA, softmax output is used to attribute the category values.

Sharp and Smooth Gradient Ascent Methods

For MFABA, two gradient ascent methods are applied namely sharp and smooth gradient ascent methods in Eq. 12. Smooth method truncates the gradients, causing a relatively weak sample gradient to traverse a same distance as a strong sample gradient. For example, a pixel with 0.01 gradient will change by the same magnitude as a pixel with 0.71 gradient. Sharp method maximises the directionality of the preserved gradient in favour of the more dominant gradient information, and the attribution results in the sharpest information.

$$\begin{aligned} smooth(grad) &= sign(grad) \\ sharp(grad) &= \frac{\text{grad}}{\| \text{ grad } \|_2} \end{aligned} \tag{12}$$

The Role of the $sign$ Function

In MFABA, the gradient ascent method utilizes the adversarial attack to find samples and the equivalent gradient direction. Normally, the gradient is calculated as $\frac{\partial F_i}{\partial x_i}$. While the adversarial attack usually chooses a loss function as the objective, the gradient will be $\frac{\partial L}{\partial x} = \frac{\partial L}{\partial F_i} \frac{\partial F_i}{\partial x} = -\frac{1}{F_i} \frac{\partial F_i}{\partial x}$, where $\frac{1}{F_i}$ only affects the vector norm not the gradient direction, and can be interpreted as an equivalence relationship with the sharp and smooth methods. In classical adversarial attack task, $sign$ function (Eq. 12) is required to prevent biased training towards the direction of larger weight (Goodfellow, Shlens, and Szegedy 2022), preventing meaningless input changes for attribution results.

In-Depth Analysis

Analysis of MFABA Efficiency

We evaluate the method based on the inference speed and the number of forward and backward propagations in a unified GPU environment. The corresponding gradient information of $\frac{\partial L(x_j)}{\partial x_j}$ is kept during the gradient ascending process, which avoid recomputing for the subsequent steps. Typically, it takes 3-10 steps to find an adversarial sample. In comparison with IG which also meets the axioms, IG requires 30-200 rounds of forward propagation between x_0 and x_n while our algorithm only requires 3-10 rounds of forward and backward propagation.

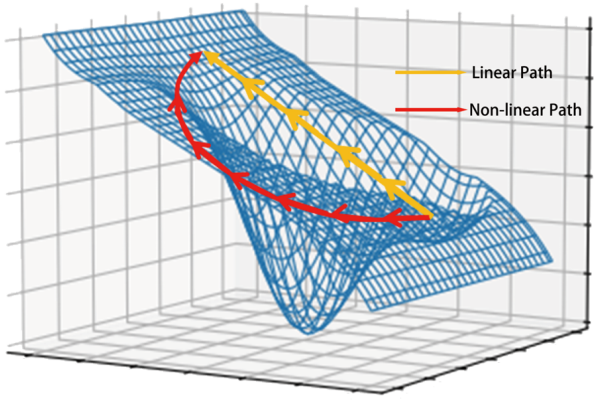


Figure 1: Linear and non-linear path of the direction for adversarial attack and baseline samples

Different from the pairwise attack approach used by BIG, MFABA algorithm does not use boundary search, instead using gradient ascent method to identify the decision boundary with adversarial samples. With Eq. 11, the samples obtained from each iteration are relatively close to each other, leading to high-quality samples for decision boundary. Other

works may specify a directed adversarial sampling attack to find the attribution results, such as (Pan, Li, and Zhu 2021). MFABA does not specific a direction of the adversarial attack to find a sample. The reason is that, for example, given a correct label of A, the adjacent decision boundary is defined as B. However, C is closer to B. In this case, a directed adversarial attack to C will not help to find the decision boundary for A correctly. Also, directly using the adversarial attack to find decision boundary would result in extensive gradient computation and adversarial sampling time. In MFABA, we avoid this by preserving the gradient graph for forward and backward propagation.

Definition of Aggressiveness

Definition: When x' satisfies the attack

$$L\left(x'\right) > L(x) \tag{13}$$

$$\left\|x - x'\right\|_p < \varepsilon \tag{14}$$

If a lower value of loss function L indicates a better performance for neural network, hereby we define the aggressiveness for x', meaning that x' is an sample with aggressiveness for L function. In other way, the sample of x' not subject to the equations is called a non-aggressive sample.

In Figure 1, the red line denotes the direction of attack while yellow line represents the baseline direction, such as the linear path in BIG. As shown in the diagram, there may exist many non-aggressive samples. BIG algorithm needs to compute $\int_0^1 \frac{\partial f((x-x_b)t+x_b)}{\partial x} dt$ in the process of attribution, changing x_b to x will result in many samples that do not have non-aggressive samples entering the computational process, and the process of non-aggressive sample computation will affect the correct attribution of the features. As shown in Figure. 2, in the process of MFABA calculation, we remove the non-aggressive sample features and visualize the aggressive or non-aggressive sample integration process separately. More details can be found in Appendix folder in GitHub link. We find that all non-aggressive samples deviated from the key features have critical impacts on the results. We will further investigate this in the next section.

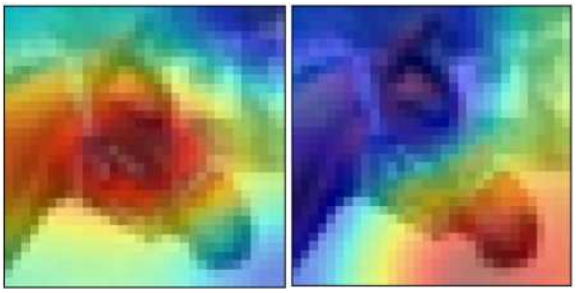


Figure 2: Comparison of heatmap without (left) and with (right) non-aggressive samples

Comparison With Other State-of-the-Art Methods

In this section, we will derive a linear approximate version of MFABA, and compare it with other methods, including IG and BIG. In BIG (Wang, Fredrikson, and Datta 2021), Eq. 15 shows that the computation can be seen as calculating the difference between x and x' and the definite integral of the computed gradient in the linear path, respectively.

$$g_{BIG}(x; x_b) = (x - x_b) \int_0^1 \frac{\partial f((x - x_b)t + x_b)}{\partial x} dt \tag{15}$$

Based on the discussion in Definition of Aggressiveness, we notice that the linear path may not be optimal. In MFABA, the non-aggressive samples are removed from the linear path. Regarding the overall computational tasks, the linear path requires more time to obtain the corresponding gradient information, at least 30-200 rounds in BIG for forward and backward propagation. At this point, we consider mapping all the attack samples generated in the adversarial attack to the linear distance, and approximating the integration result using IG algorithm.

$$t_j = \frac{\sum_{i=1}^{j} \left\|x_i - x_{i-1}\right\|_p}{\sum_{k=1}^{n} \left\|x_k - x_{k-1}\right\|_p} \tag{16}$$

$$t_j = \frac{(x_j - x_0) \cdot \cos < x_j - x_0, x_n - x_0 >}{x_n - x_0} \tag{17}$$

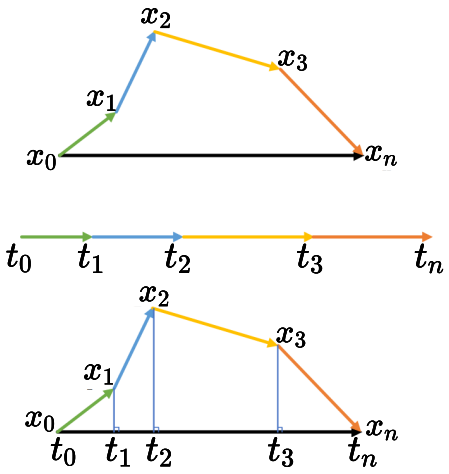


Figure 3: Two approximate algorithms of MFABA. The top graph represents the MFABA-norm algorithm, and the bottom graph represents the MFABA-cosine algorithm

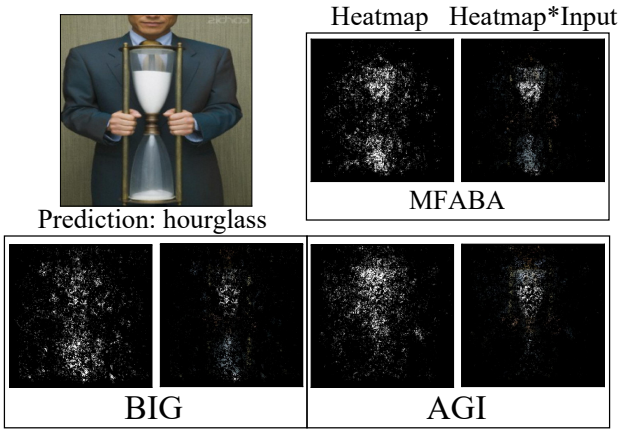
Eq. 16 is MFABA-norm and Eq. 17 is MFABA-cosine method. Figure. 3 illustrates the two approximate algorithms of MFABA for Eq. 16 and 17, respectively. Eq. 16 treats all the Lp-parametric distances of the motion trajectories x_0-x_n as 1, and the position of j-th sample between 0 and 1 can be regarded as the relative distance of the position traveled by the x_j sample. The second approach in Eq. 17 maps the relative positions of x_j and x_0 onto the vector x_0-x_n, which also reaches the objective of obtaining the relative distance. We have included more visualization results of MFABA-norm and MFABA-cosine in the Appendix folder.

Evaluation

In this section, we provide the experiment design details and the experimental results to answer following questions: 1) Can MFABA provide enhanced and faithful interpretations of the model results in comparison with other state-of-the-art methods? 2) How much improvement has been achieved for MFABA in terms of the computational efficiency?

Experiment Setup

To fairly evaluate MFABA and other state-of-the-art methods, we have implemented the experiments with publicly available and widely used model architecture including Resnet50 and EfficientNet. The empirical experiments are conducted against the CIFAR10 (Krizhevsky, Nair, and Hinton 2010), CIFAR100 and ImageNet (Russakovsky et al. 2015) datasets. It is acknowledged that the sizes of these datasets sizes exceed ten thousand. For each method, 50 gradient ascending steps are set as the maximum steps for attacking, and the learning rate is set to 0.01.



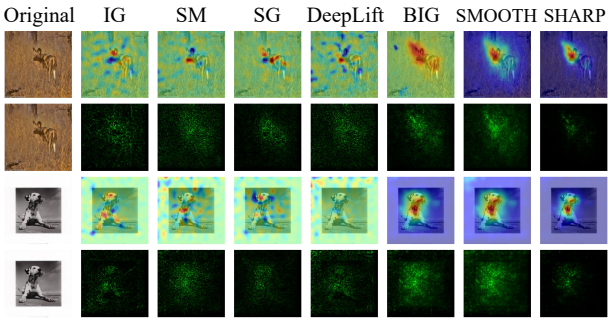


Figure 4: Results of MFABA compared to other SOTA methods (the colormaps demonstrate that our method can effectively highlight more concentrated regions associated with the recognized subjects, signifying higher interpretability)

Empirical Evaluation

Figure. 4 shows the results of the heatmap and the attribution map for different methods, including Integrated Gradients (IG), saliency map (SM), smoothed gradient (SG), DeepLift (Shrikumar, Greenside, and Kundaje 2017), BIG (Wang, Fredrikson, and Datta 2021), sharp and smooth gradient ascent methods based MFABA. It can be observed that, for MFABA results, the highlight areas are more focused related to the identified subjects, which can provide a better interpretation output. We have also provided more qualitative results in the Appendix folder.

Attribution Performance Evaluation

In addition to the visualised results for interpretation, herein we provide the statistical results for attribution performance evaluation, which are defined as the error rate evaluation indicators, the insertion and deletion score (Petsiuk, Das, and Saenko 2018) and area under accuracy information curve (Kapishnikov et al. 2019).

Error Rate Evaluation Indicators

$$ErrorRate = \left| 1 - \frac{\sum attr\left(x^i\right)}{L\left(x_n\right) - L\left(x_0\right)} \right| \tag{18}$$

The error rate is obtained by dividing the final attribution result by the true attribution sum $L\left(x_n\right) - L\left(x_0\right)$. We compare MFABA with its vanilla variant, which only utilises the first-order Taylor expansion for gradient information. We denote this method as 'Vanilla' in Table 1. We can see that the error rate of the MFABA is significantly reduced from Table 1. Overall, the error rate is relatively low, which demonstrate the high efficiency of MFABA method. A detailed discussion can be found in the Appendix folder.

Dataset	Model	Method	Error Rate
ImageNet	EfficientNet	Vanilla	0.02058
		MFABA	0.01165
CIFAR10	ResNet-50	Vanilla	0.2613
		MFABA	0.01165
CIFAR100	ResNet-50	Vanilla	0.10221
		MFABA	0.04192

Table 1: Attribution Error Rate Results

Insertion Score and Deletion Score Figure. 5 shows a schematic of our MFABA algorithm for the insertion and deletion scores. A higher insertion score corresponds to a more pronounced contribution of the input feature to the classification outcome. Conversely, a lower deletion score indicates an enhanced contribution of the input feature to the result. Our method exhibits a notable performance benefits on Inception-v3, closely trailing AGI on ResNet-50 and VGG-16. Since MFABA algorithm achieves a significant speed acceleration, we deem the trade-off to be both reasonable and acceptable.

Area Under Accuracy Information Curve In Table. 2, the employed evaluation criterion is the Area Under the Curve (AUC) of the Accuracy Information. This metric serves as an assessment tool to gauge the performance of interpretable algorithms with regard to the predictive accuracy of the model. It is evident that our method has achieved the most favorable outcomes among all the competing methods.

Model	Method	Insertion score	Deletion score	AUC
Inception-v3	SM	0.2792	0.0445	0.5150
Inception-v3	IG	0.3215	0.0445	0.5180
Inception-v3	BIG	0.4840	0.0557	0.5200
Inception-v3	AGI	0.4629	0.0590	0.5178
Inception-v3	SMOOTH	0.5368	0.0640	0.5389
Inception-v3	SHARP	0.5407	0.0627	0.5367
ResNet-50	SM	0.1441	0.0387	0.4714
ResNet-50	IG	0.1467	0.0302	0.4823
ResNet-50	BIG	0.2911	0.0485	0.4759
ResNet-50	AGI	0.3695	0.0383	0.4772
ResNet-50	SMOOTH	0.3211	0.0574	0.4854
ResNet-50	SHARP	0.3237	0.0566	0.4857
VGG16	SM	0.1018	0.0297	0.4257
VGG16	IG	0.0973	0.0249	0.4431
VGG16	BIG	0.2274	0.0390	0.4356
VGG16	AGI	0.2910	0.0320	0.4359
VGG16	SMOOTH	0.2808	0.0424	0.4540
VGG16	SHARP	0.2856	0.0410	0.4540

Table 2: Insertion score (the higher the better), deletion score (the lower the better), and AUC (the higher the better), SMOOTH and SHARP are variants of our Method MFABA

FPS Results

We use FPS, which refers to the number of frames per second (FPS) processed by the algorithms, to evaluate the algorithm processing speed. The hardware for our experiment includes: RTX 3090(24GB)*1 for GPU, 24 vCPU AMD EPYC 7642 48-Core for CPU and 80GB RAM. To comprehensively evaluate the algorithm efficiency, we have conducted two separate tests including single image testing and multiple images testing.

The single image test is the calculation time for one image at a time. An average result is obtained against the experiment datasets. For the multiple images testing, we count the maximum number of images being processed per second at the same time with the same specified hardware conditions. We have run the experiment for three times, and the average value is returned. We observed that, MFABA requires a smaller shared memory in GPU environment. Thus, with MFABA, we are able to deploy a larger batch size of images for attribution computation.

Dataset	Method	BIG	IG	AGI	MFABA
CIFAR10	ResNet-50	1.35	24.64	0.14	136.96
CIFAR100	ResNet-50	1.36	24.29	0.15	162.74
ImageNet	ResNet-50	0.37	6.76	0.28	51.58
ImageNet	EfficientNet	0.32	10.42	0.21	39.97

Table 3: FPS Results of BIG, IG, AGI, MFABA Algorithms

In Table 3, the results for multiple images testing is presented. MFABA has demonstrated its superiority over the other state-of-the-art methods, achieving a speed increase of more than 101 times compared to BIG (for CIFAR10 dataset with ResNet-50 model), and up to 139.26 times faster than

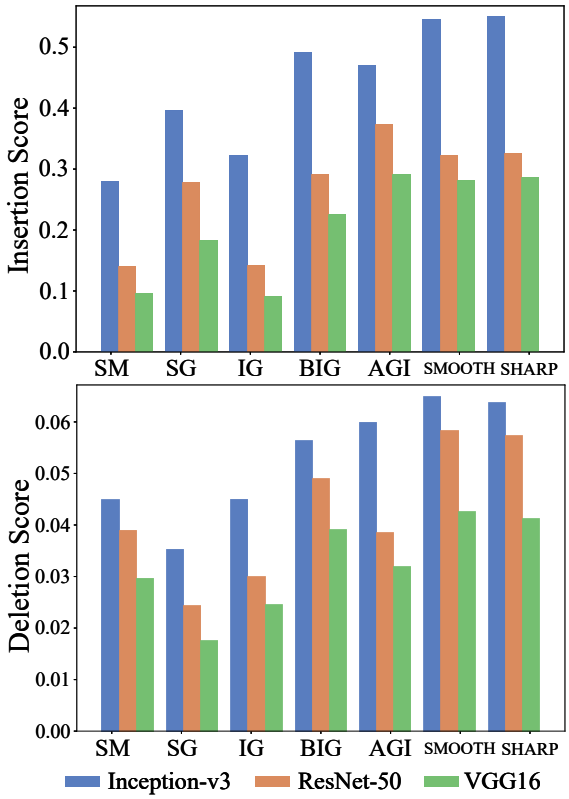


Figure 5: Insertion and Deletion score

BIG (for ImageNet dataset with ResNet-50). In comparison with IG (for ImageNet with EfficientNet), MFABA achieves near 4 times speed increase, and near 8 times faster for ImageNet dataset with ResNet-50. For AGI method, MFABA is at least 181.25 times faster for ImageNet with ResNet-50, and over 1000 times for CIFAR100 with ResNet-50. Overall, MFABA has achieved the best performance in all categories of the experiments with the datasets of CIFAR10, CIFAR100 and ImageNet.

Conclusion

In this paper, we present MFABA algorithm, a more faithful and accelerated attribution algorithm for deep neural networks interpretation. It includes two versions: sharp and smooth. We also provide the proof for the axiomatic derivation process for MFABA, which supports the two fundamental axioms of Sensitivity and Implementation Invariance. A large scale experiment shows the state-of-the-art performance of MFABA. In addition, we provide an in-depth analysis and experimental evidence, highlighting how aggressive samples can substantially contribute to the attribution output. The complete replication package is open-sourced, and we hope it will contribute to future research in advancing trustworthy AI. It is important to note, however, that our evaluation is currently limited to the conventional image dataset, omitting more intricate image tasks. Future efforts will encompass the application of MFABA in various scenarios to comprehensively assess the performance.

References

Ancona, M.; Ceolini, E.; Öztireli, C.; and Gross, M. 2017. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*.

Choi, J.; Choi, J.; and Rhee, W. 2020. Interpreting neural ranking models using grad-cam. *arXiv preprint arXiv:2005.05768*.

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*, 3(7): 620–631.

Goodfellow, I.; Shlens, J.; and Szegedy, C. 2022. Explaining and harnessing adversarial examples. arXiv [Preprint](2014). 10.48550. *arXiv preprint arXiv:1412.6572*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Janik, A.; Sankaran, K.; and Ortiz, A. 2019. Interpreting black-box semantic segmentation models in remote sensing applications.

Jeon, G.; Jeong, H.; and Choi, J. 2022. Distilled gradient aggregation: Purify features for input attribution in the deep neural network. *Advances in Neural Information Processing Systems*, 35: 26478–26491.

Jin, Z.; Zhu, Z.; Wang, X.; Zhang, J.; Shen, J.; and Chen, H. 2023. DANAA: Towards transferable attacks with double adversarial neuron attribution. In *International Conference on Advanced Data Mining and Applications*, 456–470. Springer.

Kapishnikov, A.; Bolukbasi, T.; Viégas, F.; and Terry, M. 2019. Xrai: Better attributions through regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4948–4957.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5050–5058.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2010. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4): 1.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, 99–112. Chapman and Hall/CRC.

Li, Y. 2022. Research and application of deep learning in image recognition. In *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, 994–999. IEEE.

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

Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; and Gao, J. 2021. Deep learning–based text classification: a comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3): 1–40.

Mo, Y.; Wu, Y.; Yang, X.; Liu, F.; and Liao, Y. 2022. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493: 626–646.

Pan, D.; Li, X.; and Zhu, D. 2021. Explaining Deep Neural Network Models with Adversarial Gradient Integration. In *IJCAI*, 2876–2883.

Patra, A.; and Noble, J. A. 2020. Incremental learning of fetal heart anatomies using interpretable saliency maps. In *Annual Conference on Medical Image Understanding and Analysis*, 129–141. Springer.

Petsiuk, V.; Das, A.; and Saenko, K. 2018. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*.

Ramaswamy, H. G.; et al. 2020. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 983–991.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

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

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Srinivas, S.; and Fleuret, F. 2019. Full-gradient representation for neural network visualization. *Advances in neural information processing systems*, 32.

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

Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; and Hu, X. 2020. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 24–25.

Wang, Z.; Fredrikson, M.; and Datta, A. 2021. Robust models are more interpretable because attributions look normal. *arXiv preprint arXiv:2103.11257*.

Zaidi, S. S. A.; Ansari, M. S.; Aslam, A.; Kanwal, N.; Asghar, M.; and Lee, B. 2022. A survey of modern deep learning based object detection models. *Digital Signal Processing*, 103514.

Zhang, Q.-s.; and Zhu, S.-C. 2018. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1): 27–39.

Zhang, Y.; Wang, X.; and Tang, H. 2019. An improved Elman neural network with piecewise weighted gradient for time series prediction. *Neurocomputing*, 359: 199–208.