Unifying Fourteen Post-Hoc Attribution Methods With Taylor Interactions

Huiqi Deng, Na Zou, Mengnan Du, Weifu Chen, Guocan Feng, Ziwei Yang, Zheyang Li, and Quanshi Zhang, *Member, IEEE*

***Abstract*—Various attribution methods have been developed to explain deep neural networks (DNNs) by inferring the attribution/importance/contribution score of each input variable to the final output. However, existing attribution methods are often built upon different heuristics. There remains a lack of a unified theoretical understanding of why these methods are effective and how they are related. Furthermore, there is still no universally accepted criterion to compare whether one attribution method is preferable over another. In this paper, we resort to Taylor interactions and for the first time, we discover that fourteen existing attribution methods, which define attributions based on fully different heuristics, actually share the same core mechanism. Specifically, we prove that attribution scores of input variables estimated by the fourteen attribution methods can all be mathematically reformulated as a weighted allocation of two typical types of effects, *i.e.*, independent effects of each input variable and interaction effects between input variables. The essential difference among these attribution methods lies in the weights of allocating different effects. Inspired by these insights, we propose three principles for fairly allocating the effects, which serve as new criteria to evaluate the faithfulness of attribution methods. In summary, this study can be considered as a new unified perspective to revisit fourteen attribution methods, which theoretically clarifies essential similarities and differences among these methods. Besides, the proposed new principles enable people to make a direct and fair comparison among different methods under the unified perspective.**

***Index Terms*—Attribution methods, Taylor interactions.**

I. Introduction

DESPITE widespread success in a variety of real-world applications, DNNs are typically regarded as "black boxes", because it is difficult to interpret how a DNN makes a decision. The lack of interpretability hampers their wide applications on high-stake tasks, such as automatic driving [9] and AI healthcare [26]. Therefore, interpreting DNNs has drawn increasing attentions recently.

As a typical perspective of interpreting DNNs, attribution methods aim to calculate the attribution/importance/contribution score of each input variable to the network output [10], [22], [29]. For example, given a pre-trained DNN for image classification and an input image, the attribution score of each input variable refers to the numerical effect of each pixel on the confidence score of classification.

Although many attribution methods have been proposed in recent years [5], [10], [29], most of them are built upon different heuristics. For example, some methods [33], [37] consider that the gradient of the output *w.r.t.* the input can reflect the importance of input variables. In addition, some methods [40], [41] use the output change when the input variable x_i is occluded to measure the importance.

There is *a lack of unified theoretical perspective* to examine the correctness of these attribution methods, or at least to mathematically clarify their core mechanisms, *e.g.*, explaining their essential similarity and difference, and comparing their advantages and disadvantages. A few researchers have attempted to unify different attribution methods [2][29] [22], but these studies cover only a few methods (please see Table I for details).

In this paper, we propose the Taylor interaction as a new unified perspective, which first allows us to mathematically formulate mechanisms of up to fourteen attribution methods into the same system. Furthermore, the fourteen attribution methods cover different types of attribution methods that are based on fully different heuristic designs, including gradient-based methods, back-propagation methods, and perturbation-based methods. We believe that a mathematical system that unifies various methods is more likely to reflect essential factors in generating attributions, and enables a fair comparison between

Manuscript received 26 February 2023; revised 16 December 2023; accepted 17 January 2024. Date of publication 25 January 2024; date of current version 5 June 2024. This work was supported in part by the National Nature Science Foundation of China under Grant 92370115, in part by the National Nature Science Foundation of China under Grant 62276165, in part by the National Key R&D Program of China under Grant 2021ZD0111602, in part by Shanghai Natural Science Foundation under Grants 21JC1403800 and 21ZR1434600, and in part the CCF-Hikvision Open fund. The work was done under the supervision of Dr. Quanshi Zhang. Recommended for acceptance by M.-M. Cheng. *(Corresponding author: Quanshi Zhang.)*

Huiqi Deng is with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: denghq7@sjtu.edu.cn).

Na Zou is with the Department of Industrial Engineering, University of Houston, Houston, TX 77204 USA (e-mail: nzou1@tamu.edu).

Mengnan Du is with the Department of Data Science, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: mengnan.du@njit.edu).

Weifu Chen is with the Department of Computer Science, Guangzhou Maritime University, Guangzhou, Guandong 510725, China (e-mail: waifook.chan@gmail.com).

Guocan Feng is with the School of Mathematics and Computational Science, Sun Yat-sen University, Guangzhou, Guangdong 510275, China (e-mail: mcsfgc@mail.sysu.edu.cn).

Ziwei Yang and Zheyang Li are with the Hikvision Research Institute, Hangzhou, Zhejiang 310051, China (e-mail: yangziwei5@hikvision.com; lizheyang@hikvision.com).

Quanshi Zhang is with the Department of Computer Science and Engineering, and the John Hopcroft Center, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zqs1022@sjtu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3358410, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3358410

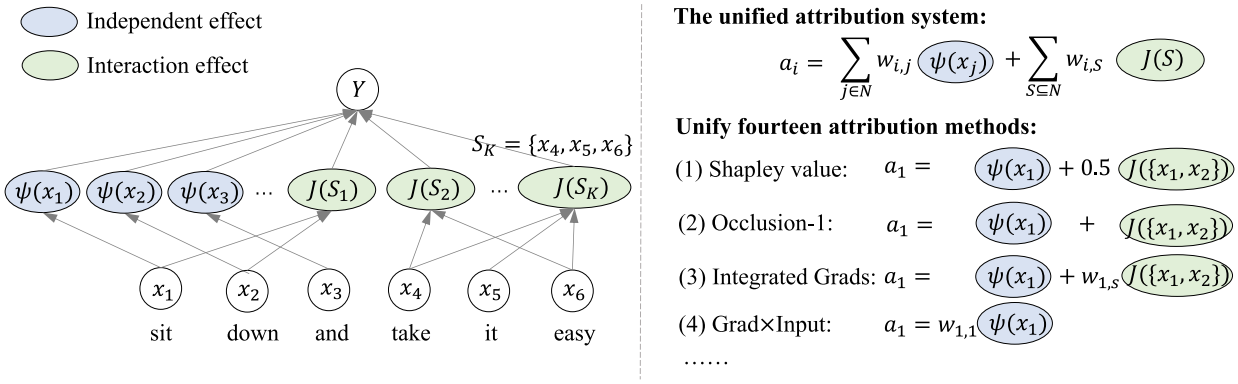


Fig. 1. (Left) We prove that the network output Y can be mathematically decomposed as the sum of two typical types of effects caused by input variables, i.e., independent effects $\psi(x_i)$ of each input variable x_i and interaction effects $J(S)$ between input variables in each subset S. (Right) In this paper, we prove a unified explanation for fourteen attribution methods, although these methods are designed based on different heuristics. That is, the attribution score a_i of each variable x_i in each method can all be reformulated as a weighted allocation of independent effects $\{\psi(x_j)\}_{j \in N}$ and interaction effects $\{J(S)\}_{S \subseteq N}$. Here, $w_{i,j}$ and $w_{i,S}$ denote the corresponding weights for allocating effects. The essential difference among these methods lies in different weights of allocation. .

TABLE I
SUMMARY OF WORKS ON UNIFYING ATTRIBUTION METHODS

Work	Unification	# methods
[22]	Additive feature attribution	6
[2]	Modified gradient × input	5
[29]	First-order Taylor framework	4
Ours	Taylor interaction perspective	14

different attribution methods. An overview of the proposed Taylor interaction perspective is illustrated in Fig. 1.

The proposed Taylor interaction is a new metric to represent the two types of effects on the network output caused by input variables. First, an input variable may make a direct effect on the network output, which is not influenced by other input variables. Such an effect is termed an *independent effect*. Second, an input variable may also collaborate with other input variables to affect the network output. Such an effect is termed an *interaction effect*. Both types of effects can be quantified as specific Taylor interactions.

As a toy example, let us consider a DNN for a scene classification task that is trained to fit the target function $f_{\text{study room}}(\boldsymbol{x}) = 3x_{\text{book}} + 2x_{\text{desk}}x_{\text{lamp}}x_{\text{book}}$. Here, the binary input variables $x_{\text{book}}, x_{\text{desk}}, x_{\text{lamp}} \in \{0, 1\}$ denote the present/absent state of these objects in the scene. The book variable x_{book} has an independent effect $3x_{\text{book}}$ on the output. Collaboration between variables $x_{\text{desk}}, x_{\text{lamp}}, x_{\text{book}}$ has an interaction effect $2x_{\text{desk}}x_{\text{lamp}}x_{\text{book}}$ on the scene classification.

In this paper, we prove that attributions scores generated by fourteen different attribution methods can *all* be explained by the above two types of effects. The *essential mechanism* of each attribution method can be mathematically represented as allocating a specific ratio of each independent effect and a specific ratio of each interaction effect to the input variable x_i, so as to compute the attribution score of x_i. Furthermore, the *essential difference* between these attribution methods is that they compute attribution scores by allocating different ratios of independent effects and interaction effects to input variables.

To understand the above unified perspective, let us consider the previous example $f_{\text{study room}}(\boldsymbol{x}) = 3x_{\text{book}} + 2x_{\text{desk}}x_{\text{lamp}}x_{\text{book}}$. There are an independent effect $3x_{\text{book}}$ and an interaction effect $2x_{\text{desk}}x_{\text{lamp}}x_{\text{book}}$. Then, attribution scores of the variable x_{book} in fourteen attribution methods can all be represented as $a_{\text{book}} = w_1 \cdot 3x_{\text{book}} + w_2 \cdot 2x_{\text{desk}}x_{\text{lamp}}x_{\text{book}}$, where w_1, w_2 are ratios of allocating independent effects and interaction effects, respectively. These attribution methods differ in how to determine the ratios w_1 and w_2. For example, the Shapley value [22] allocates the entire independent effect to the variable x_{book}, and allocate 1/3 of the interaction effect to x_{book}. In this way, the attribution is computed as $a_{\text{book}} = 1 \cdot 3x_{\text{book}} + 1/3 \cdot 2x_{\text{desk}}x_{\text{lamp}}x_{\text{book}}$. In comparison, the Occlusion-1 [40] allocates the entire independent effect and the entire interaction effect to the variable x_{book}. That is, the attribution is $a_{\text{book}} = 1 \cdot 3x_{\text{book}} + 1 \cdot 2x_{\text{desk}}x_{\text{lamp}}x_{\text{book}}$.

Principles of faithful attribution. The above unified perspective enables us to directly compare different attribution methods. To this end, we propose three principles to examine whether one attribution method fairly allocates the two types of effects to input variables, which serve as a criterion to evaluate the faithfulness of the attribution method. Let us use the previous example to explain the three principles.

(i) The independent effect of a variable ($3x_{\text{book}}$) is directly caused by the variable (x_{book}), which is not influenced by other variables. Therefore, the independent effect ($3x_{\text{book}}$) is supposed to be allocated entirely to the variable (x_{book}). Other variables should not be allocated such an effect.

(ii) The interaction effect ($2x_{\text{desk}}x_{\text{lamp}}x_{\text{book}}$) is caused by the collaboration between its own set of variables ($S = \{x_{\text{desk}}, x_{\text{lamp}}, x_{\text{book}}\}$). Hence, such an effect is supposed to be allocated to involved variables, not to variables without participating in the collaboration.

(iii) The interaction effect ($2x_{\text{desk}}x_{\text{lamp}}x_{\text{book}}$) should be *all* allocated to involved variables. In other words, when we sum up the numerical effects allocated to involved variables, we obtain the exact value of the overall interaction effect.

Subsequently, we apply the three principles to evaluate the faithfulness of the fourteen attribution methods. We find that attribution methods such as Shapley value [22], Integrated Gradients [37], and DeepLIFT Rescale [32] satisfy all principles.

In summary, this paper has three contributions:

- We propose the Taylor interaction as a new unified perspective to theoretically explain the core mechanism of fourteen attribution methods.
- For each specific attribution method, the unified perspective enables us to clarify its distinctive property of computing attributions.
- We propose three principles to evaluate the faithfulness of an attribution method, which evaluate whether the method fairly allocates independent effects and interaction effects to input variables.

Significant improvements over our previous conference paper. The preliminary version of this paper has been published in AAAI [6], where we proposed a unified Taylor system to explain and unify seven attribution methods. In comparison, the current paper makes several significant improvements over [6]: (i) We have formally provided the exact definition of the Taylor interaction in Section III-A. In the AAAI paper, we only used a specific Taylor interaction effect in the second-order Taylor expansion as an intuitive example for illustration. (ii) We have proved seven new Theorems in Section III-D to reformulate and unify another seven attribution methods into our system, including *Prediction Difference*, *Grad-CAM*, *Shapley value*, *LRP-*$\alpha\beta$, *Deep Taylor*, *DeepLIFT RevealCancel*, and *Deep Shap*. (iii) We have proved a close connection between our Taylor interactions and typical game-theoretic interactions in Section III-B, which indicates the generality and solidness of Taylor interactions. (iv) We have conducted new experiments to visualize more attribution results, verify the correctness of our theory, and examine the fidelity of the proposed principles in evaluating attribution methods.

II. Related Work

A. Existing Attribution Methods

Various attribution methods have been developed to interpret DNNs, which infer the contribution score of each input variable to the output. In general, existing attribution methods can be roughly categorized into the following three types.

Gradient-based attribution methods. The *Gradient* method [4] considers the gradient of the network output *w.r.t.* each input variable as the attribution of the input variable. The *Gradient* $\times$ *Input* method [33] formulates attributions as the element-wise product of gradients and input features. The *Integrated Gradients* method [37] estimates attributions as the element-wise product of the average gradient and input, where gradients are averaged when the input varies along a linear path from the input sample to a baseline point. The *Expected Gradients* method [12] averages attribution results estimated by *Integrated Gradients* over multiple baseline points. Besides, the *Grad-CAM* method [31] uses the average gradient of the loss *w.r.t.* all features in a channel as the weight for the channel, and uses such a channel-wise weight to compute the attribution score over different locations.

Back-propagation attribution methods estimate attributions of intermediate features at a layer and then recursiviely back-propagate these attributions to the previous layer, until obtaining the attribution scores of variables in the input layer. This type of method includes *LRP-*ϵ [3], *LRP-*$\alpha\beta$ [3], *Deep Taylor* [23], *DeepLIFT Rescale* [32], *Deep SHAP* [22], *DeepLIFT RevealCancel* [32], and so on [5]. The essential difference between different back-propagation methods is that they employ different recursive rules for back-propagating attributions between two adjacent layers, which is detailedly introduced in Section III-C.

Perturbation-based attribution methods infer the attribution of an input variable according to how much masking the variable will affect the network output. The *Occlusion-1* method [40] and the *Occlusion-patch* method [41] formulate the attribution of a pixel (patch) as the output change when the pixel (patch) is unmasked *w.r.t.* the case when the pixel (patch) is masked. Moreover, the *Shapley value* method [22] estimates the attribution by averaging such output changes when masking states of other variables vary. In addition, several attribution methods [5], [14], [15], [16] seek a subset of the most influential input variables whose masking will cause the most significant change in the network output.

In this paper, we explain and unify the mechanisms of as many as fourteen existing attribution methods, which cover most mainstream attribution methods.

B. Understand and Unify Attribution Methods

There are a few works on theoretically understanding mechanisms of existing heuristic attribution methods. For example, the *Deconvnet* method [40] and the *GBP* method [35] have been theoretically proved to essentially construct (partial) recovery to the input [25], which is unrelated to the decision-making process. Besides, some efforts have also been devoted to unifying various attribution methods. For example, *LIME* [28], *LRP-*ϵ [3], *DeepLIFT* [32], and *Shapley value* [22] are unified under the framework of additive feature attribution [22]. Some attribution methods including *Gradient* $\times$ *Input* [33], *LRP-*ϵ [3], *DeepLIFT* [32] and *Integrated Gradients* [37], are unified as multiplying a modified gradient with the input [2]. In addition, [24], [29] have shown that the attributions generated by *LRP-*ϵ [3] and *LRP-*$\alpha\beta$ [3] could be reformulated as the first-order Taylor decomposition. To the best of our knowledge, our research is the first work to leverage Taylor interaction effects to formally define the attribution problem and unify up to fourteen existing attribution methods.

III. Unifying Attribution Methods

Attribution methods have been developed as a typical perspective of explaining DNNs [10] [22], which infer the attribution/importance/contribution score of each input variable (*e.g.*, an image pixel, a word) to the final output. Specifically, given a pre-trained DNN f and an input sample $\boldsymbol{x} = [x_1, \dots x_n]^T \in \mathbb{R}^n$, an attribution method estimates an attribution vector $\boldsymbol{a} = [a_1, \dots, a_n]^T \in \mathbb{R}^n$, where a_i denotes the numerical effect of the input variable i on a scalar output of the DNN $f(\boldsymbol{x}) \in \mathbb{R}$. For example, in the classification task, $f(\boldsymbol{x})$ can be set as the classification probability of the target category. For convenience,

TABLE II
NOTATION IN THIS PAPER

Notation	Description
f	pre-trained DNN
$\boldsymbol{x}$	input sample $[x_1, \cdots, x_n]^T$
$\boldsymbol{b}$	baseline point $[b_1, \cdots, b_n]^T$
$\boldsymbol{a}$	attribution vector $[a_1, \cdots, a_n]^T$
N	index set of input variables $\{1, \cdots, n\}$
S	subset of N, $S \subseteq N$
$\boldsymbol{\kappa}$	degree vector in a Taylor expansion term
$I(\boldsymbol{\kappa})$	Taylor interaction effect
$\phi(\boldsymbol{\kappa})$	Taylor independent effect
$S_{\boldsymbol{\kappa}}$	variables involving in the interaction $I(\boldsymbol{\kappa})$
Ω_i	set of degree vectors $\boldsymbol{\kappa}$, *s.t.* $S_{\boldsymbol{\kappa}} = \{i\}$
Ω_S	set of degree vectors $\boldsymbol{\kappa}$, *s.t.* $S_{\boldsymbol{\kappa}} = S$
Ω_{ind}	$\cup_{i \in N} \Omega_i$, refers to all independent effects
Ω_{int}	$\cup_{S \subseteq N, \lvert S \rvert > 1} \Omega_S$, refers to all interaction effects
$\psi(i)$	generic independent effect of the variable i
$J(S)$	generic interaction effect of variables in S

we summary all descriptions of main notations for this paper in Table II.

Although various attribution methods have been proposed recently, most of them are built upon different heuristics. There still lacks a unified perspective to explain why these attribution methods are effective and how they are related. Therefore, in this paper, we propose the Taylor interaction as a new unified perspective, which allows us to explain the mechanisms of up to fourteen attribution methods.

A. Explaining a DNN by Taylor Interaction Effects

In this subsection, we propose the Taylor interaction as a new perspective, which mathematically proves that the output of a DNN can be decomposed into two typical types of effects, including the Taylor independent effect of each input variable and the Taylor interaction effect between input variables. In the following subsections, we will use the two effects to explain and compare the core mechanisms of different attribution methods.

Preliminaries: Taylor expansion of a DNN. Given a pre-trained DNN f and an input sample $\boldsymbol{x} = [x_1, \ldots, x_n]^T$ with n input variables (indexed by $N = \{1, \ldots, n\}$), let us consider the K-order Taylor expansion[1] of the DNN f, which is expanded at a baseline point $\boldsymbol{b} = [b_1, \ldots, b_n]^T$.

$$\begin{aligned} f(\boldsymbol{x}) &= f(\boldsymbol{b}) + \sum_{i=1}^{n} \frac{1}{1!} \cdot \frac{\partial f(\boldsymbol{b})}{\partial x_i} \cdot (x_i - b_i) \\ &\quad + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2!} \cdot \frac{\partial^2 f(\boldsymbol{b})}{\partial x_i \partial x_j} \cdot (x_i - b_i)(x_j - b_j) + \cdots + \epsilon_K \\ &= f(\boldsymbol{b}) + \sum_{k=1}^{K} \sum_{\boldsymbol{\kappa} \in O_k} \underbrace{C(\boldsymbol{\kappa}) \cdot \triangledown f(\boldsymbol{\kappa}) \cdot \pi(\boldsymbol{\kappa})}_{\text{defined as } I(\boldsymbol{\kappa})} + \epsilon_K \end{aligned} \tag{1}$$

[1] Note that although deep networks with ReLU activation are not differentiable such that the Taylor expansion is not applicable, we can use networks with softplus activation (the approximation of ReLU) to provide insight into the rationale behind ReLU networks.

where ϵ_K denotes the approximation error of the K-order expansion. Each expansion term $I(\boldsymbol{\kappa})$ is defined as follows, which consists of the coefficient $C(\boldsymbol{\kappa})$, the partial derivative $\triangledown f(\boldsymbol{\kappa})$, and the product $\pi(\boldsymbol{\kappa})$.

$$\begin{aligned} I(\boldsymbol{\kappa}) &\overset{\text{def}}{=} C(\boldsymbol{\kappa}) \cdot \triangledown f(\boldsymbol{\kappa}) \cdot \pi(\boldsymbol{\kappa}) \\ s.t. \quad C(\boldsymbol{\kappa}) &= \frac{1}{(\kappa_1 + \cdots + \kappa_n)!} \binom{\kappa_1 + \cdots + \kappa_n}{\kappa_1, \ldots, \kappa_n} \\ \triangledown f(\boldsymbol{\kappa}) &= \frac{\partial^{\kappa_1 + \cdots + \kappa_n} f(\boldsymbol{b})}{\partial^{\kappa_1} x_1 \cdots \partial^{\kappa_n} x_n} \\ \pi(\boldsymbol{\kappa}) &= (x_1 - b_1)^{\kappa_1} \ldots (x_n - b_n)^{\kappa_n} \end{aligned} \tag{2}$$

where $\boldsymbol{\kappa} = [\kappa_1, \ldots, \kappa_n] \in \mathbb{N}^n$ denotes the degree vector of the expansion term $I(\boldsymbol{\kappa})$, and $\kappa_i \in \mathbb{N}$ denotes the non-negative integral degree of the variable i.

Moreover, we classify all expansion terms in (1) into different orders. The order of each expansion term $I(\boldsymbol{\kappa})$ is defined as its overall degree, *i.e.*, $\text{order}(I(\boldsymbol{\kappa})) = \kappa_1 + \cdots + \kappa_n$. In this way, we can use the set of degree vectors $O_k = \{\boldsymbol{\kappa} \in \mathbb{N}^n | \kappa_1 + \cdots + \kappa_n = k\}$ to represent all expansion terms of the k-th order.

Taylor interaction effects. In (1), each Taylor expansion term $I(\boldsymbol{\kappa})$ represents an interaction between input variables in the set $S_{\boldsymbol{\kappa}}$. Here, $S_{\boldsymbol{\kappa}}$ denotes the receptive field of the interaction $I(\boldsymbol{\kappa})$, *i.e.*, the set of all variables involved in the interaction.

$$S_{\boldsymbol{\kappa}} \overset{\text{def}}{=} \{i | \kappa_i > 0\} \tag{3}$$

Let us take the Taylor expansion term $I(\boldsymbol{\kappa}) = c \cdot (x_{\text{eye}} - b_{\text{eye}})^2 (x_{\text{nose}} - b_{\text{nose}})(x_{\text{mouth}} - b_{\text{mouth}})$ of the DNN for face recognition as an example, where $\kappa_{\text{eye}} = 2, \kappa_{\text{nose}} = 1$ and $\kappa_{\text{mouth}} = 1$, respectively. This expansion term $I(\boldsymbol{\kappa})$ indicates that variables in $S_{\boldsymbol{\kappa}} = \{\text{eye}, \text{nose}, \text{mouth}\}$ interact with each other to form an AND pattern. Only when all variables in $S_{\boldsymbol{\kappa}}$ co-appear, the AND pattern is formed and makes an interaction effect $I(\boldsymbol{\kappa})$ on the output $f(\boldsymbol{x})$ of the DNN. Instead, masking any of variables of $x_{\text{eye}}, x_{\text{nose}}$, and x_{mouth} using their baseline value b_i will deactivate the AND pattern and remove the numerical effect from the network output, *i.e.*, making $I(\boldsymbol{\kappa}) = 0$. Therefore, $I(\boldsymbol{\kappa})$ quantifies the effect of the interaction (AND pattern) on the network output, which is termed the *Taylor interaction effect*.

Taylor independent effects. We further define a specific type of Taylor interaction effect $I(\boldsymbol{\kappa})$, where only a single variable is involved in the interaction ($|S_{\boldsymbol{\kappa}}| = 1$), as the *Taylor independent effect*. We denote the Taylor independent effect by a new notation $\phi(\boldsymbol{\kappa})$ to differentiate it from other Taylor interaction effects.

$$\phi(\boldsymbol{\kappa}) \overset{\text{def}}{=} I(\boldsymbol{\kappa}), \quad \forall\, \boldsymbol{\kappa} \in \{\boldsymbol{\kappa} \mid |S_{\boldsymbol{\kappa}}| = 1\}. \tag{4}$$

The Taylor independent effect represents the effect of a single variable without depending on (interacting with) other variables. For example, when the degree vector $\boldsymbol{\kappa} = [0, \ldots, 0, \kappa_i, 0, \ldots, 0]$ satisfying $S_{\boldsymbol{\kappa}} = \{i\}$, the Taylor independent effect is computed as

$$\phi(\boldsymbol{\kappa}) = \frac{1}{\kappa_i!} \frac{\partial^{\kappa_i} f(\boldsymbol{b})}{\partial^{\kappa_i} x_i} (x_i - b_i)^{\kappa_i}. \tag{5}$$

TABLE III
TOY EXAMPLE TO ILLUSTRATE THE TAYLOR (GENERIC) INDEPENDENT EFFECT AND THE TAYLOR (GENERIC) INTERACTION EFFECT

Neural network	$f(\boldsymbol{x}) = x_1 + x_1^3 + x_1x_2x_3^2 + x_1^3x_2x_3^2$
Taylor independent effect	$\phi(\boldsymbol{\kappa_1}) = x_1$ with $\boldsymbol{\kappa_1} = [1,0,0]$, $\phi(\boldsymbol{\kappa_2}) = x_1^3$ with $\boldsymbol{\kappa_2} = [3,0,0]$
Taylor interaction effect	$I(\boldsymbol{\kappa_3}) = x_1x_2x_3^2$ with $\boldsymbol{\kappa_3} = [1,1,2]$, $I(\boldsymbol{\kappa_4}) = x_1^3x_2x_3^2$ with $\boldsymbol{\kappa_4} = [3,1,2]$
Generic independent effect	$\psi(x_1) = x_1 + x_1^3$
Generic interaction effect	$J(\{x_1,x_2,x_3\}) = x_1x_2x_3^2 + x_1^3x_2x_3^2$

To avoid ambiguity, in the following manuscript, we use the Taylor independent effect $\phi(\boldsymbol{\kappa})$ to represent the effect of a single variable without depending on other variables ($|S_{\boldsymbol{\kappa}}| = 1$), and use the Taylor interaction effect $I(\boldsymbol{\kappa})$ to represent the interaction effect among multiple variables ($|S_{\boldsymbol{\kappa}}| > 1$).

Decomposing the network output into the independent effect of each input variable and the interaction effect of each set of input variables. For a specific set of input variables $S(S \subseteq N, |S| > 1)$, let us consider the overall effect caused by interactions between variables in S, denoted by $J(S)$, which is defined as the summation of all Taylor interaction effects *w.r.t.* the receptive field S. We term $J(S)$ the *generic interaction effect* for S.

$$J(S) \overset{\text{def}}{=} \sum\nolimits_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa}),\ s.t.\ \Omega_S = \{\boldsymbol{\kappa}|S_{\boldsymbol{\kappa}} = S\}, \tag{6}$$

where Ω_S is the set of degree vectors $\boldsymbol{\kappa}$ corresponding to all Taylor interaction effects $I(\boldsymbol{\kappa})$ with the same receptive field S. More specifically, $\Omega_S = \{\boldsymbol{\kappa}|\ \forall i \in S, \kappa_i > 0$ and $\forall i \notin S, \kappa_i = 0\}$.

Similarly, we define the *generic independent effect* $\psi(i)$ of the variable i to measure the overall effect of the variable i without interacting with other variables.

$$\psi(i) \overset{\text{def}}{=} \sum\nolimits_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}),\ s.t.\ \Omega_i = \{\boldsymbol{\kappa}|S_{\boldsymbol{\kappa}} = \{i\}\}. \tag{7}$$

where Ω_i is the set of degree vectors $\boldsymbol{\kappa}$ corresponding to all Taylor independent effects $\phi(\boldsymbol{\kappa})$ of the variable i. More specifically, $\Omega_i = \{\boldsymbol{\kappa}|\kappa_i > 0$ and $\forall j \neq i, \kappa_j = 0\}$.We provide a toy example in Table III to illustrate the Taylor (generic) independent effect and Taylor (generic) interaction effect.

Proposition 1. (Proof in Appendix A, available online) The network output $f(\boldsymbol{x})$ can be decomposed as the sum of generic independent effects $\psi(i)$ of different variables and generic interaction effects $J(S)$ w.r.t. different subsets of variables.

$$\begin{aligned} f(\boldsymbol{x}) &= f(\boldsymbol{b}) + \sum_{i\in N}\sum_{\boldsymbol{\kappa}\in\Omega_i}\phi(\boldsymbol{\kappa}) + \sum_{S\subseteq N,|S|>1}\sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa}) \\ &= f(\boldsymbol{b}) + \sum\nolimits_{i\in N}\psi(i) + \sum\nolimits_{S\subseteq N,|S|>1} J(S) \end{aligned} \tag{8}$$

B. Connections Between the Taylor Interaction Effect and the Harsanyi Dividend

In this subsection, we prove the close theoretical connection between our Taylor interaction effect (generic interaction effect) and the typical Harsanyi dividend interaction [18], as shown in Theorem 1.

The Harsanyi dividend $H(S)$ is a typical game-theoretic interaction metric to measure the interaction effect between a specific set S of input variables, which is computed as:

$$H(S) = \sum\nolimits_{T\subseteq S}(-1)^{|T|-|S|} f(\boldsymbol{x}_T),\ \ \forall S \subseteq N, |S| > 1 \tag{9}$$

where $f(\boldsymbol{x}_T)$ denotes the network output when variables in T of the input sample $\boldsymbol{x}$ remain unchanged, and variables in $N \setminus T$ are masked using pre-defined baseline values, *i.e.*, $\forall i \in N \setminus T$, setting $x_i = b_i$.

The Harsanyi dividend is considered a general interaction metric. This is because [27] has proven that the Harsanyi dividend satisfies seven desirable axioms, and can be considered an elementary component of many existing game-theoretic interaction metrics, such as the Shapley interaction index [17] and Shapley Taylor interaction index [36].

Theorem 1. (Proof in Appendix A, available online) The Harsanyi dividend $H(S)$ is equivalent to the generic interaction effect $J(S)$ between variables in S, which is defined in (6).

$$H(S) = J(S) = \sum\nolimits_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa}),\ \ \forall S \subseteq N, |S| > 1 \tag{10}$$

Theorem 1 proves the equivalence between the typical Harsanyi dividend interaction metric and our generic interaction effect, which indicates the generality and solidness of Taylor interactions.

C. Rewriting Attributions as a Weighted Allocation of Independent Effects and Interaction Effects

In this subsection, we revisit the attribution problem from the above Taylor interaction perspective. As shown in Fig. 2, we discover that all attributions generated by fourteen different attribution methods can be unified into the same system, *i.e.*, represented as a weighted allocation of independent effects and interaction effects.

The basic idea of our unified attribution system is intuitive. As proved in Proposition 1, the Taylor interaction shows the capability to exhaustively quantify the generic independent effect $\psi(i)$ of each input variable i and all generic interaction effects $J(S)$ $(i \in S)$ caused by the input variable i's collaboration with other variables. As a result, it is natural to consider that each attribution method assigns a specific ratio of each effect to the attribution score of the variable i. In this way, we use the set of ratios for effect allocation to explain the characteristic of each attribution method.

Specifically, let a_i denote the attribution score of the variable i. In this paper, we *prove* that although fourteen existing attribution methods are designed on different heuristics, a_i estimated by each method can all be reformulated into the following paradigm, *i.e.*, represented as a specific re-allocation of generic

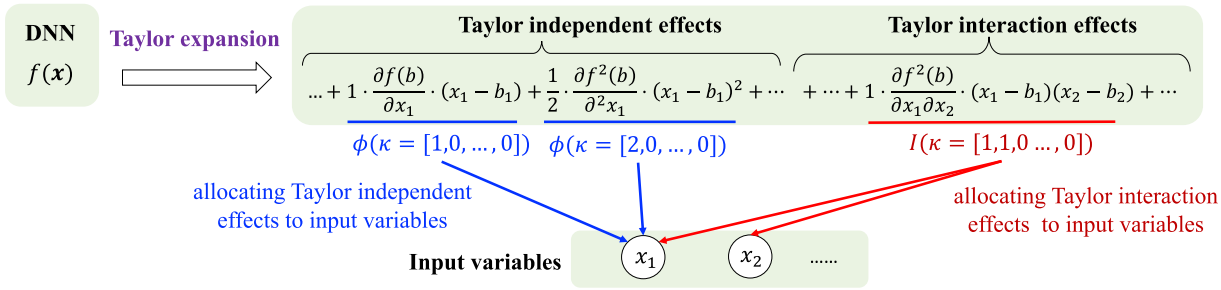


Fig. 2. Understanding and unifying attribution methods via Taylor interactions. In this paper, we prove that each attribution method is mathematically equivalent to the following flowchart, *i.e.*, each attribution method first explains the network output $f(\boldsymbol{x})$ as a Taylor expansion model, thereby decomposing the network output $f(\boldsymbol{x})$ into Taylor independent effect $\phi(\boldsymbol{\kappa})$ of each input variable x_i and the Taylor interaction effect $I(\boldsymbol{\kappa})$ between each set of input variables (introduced in Section III-A). Then, this method accordingly re-allocates the two typical types of effects to each input variable x_i, so as to compute the attribution score a_i (introduced in Section III-C). Here, we show Taylor expansion terms of only the first and second orders for simplicity. .

independent effects and generic interaction effects.

$$a_i = \sum\nolimits_{j\in N} \underbrace{w_{i,j}\cdot\psi(j)}_{\stackrel{\text{def}}{=}a_{i\leftarrow\psi(j)}} + \sum\nolimits_{S\subseteq N,|S|>1} \underbrace{w_{i,S}\cdot J(S)}_{\stackrel{\text{def}}{=}a_{i\leftarrow J(S)}} \quad (11)$$

where $w_{i,j}$ ($w_{i,S}$) denotes the ratio of j's generic independent effect $\psi(j)$ (the generic interaction effect $J(S)$ between variables in S) being allocated to the input variable i. Accordingly, we can use $a_{i\leftarrow\psi(j)} \stackrel{\text{def}}{=} w_{i,j}\psi(j)$ and $a_{i\leftarrow J(S)} \stackrel{\text{def}}{=} w_{i,S}J(S)$ to represent the allocated effects from $\psi(j)$ and $J(S)$, respectively.

To be precise, we can further expand the above (11) as a re-allocation of Taylor independent effects and Taylor interaction effects.

$$\begin{aligned} a_i &= \sum\nolimits_{\boldsymbol{\kappa}\in\Omega_{\text{ind}}} w_{i,\boldsymbol{\kappa}}\cdot\phi(\boldsymbol{\kappa}) + \sum\nolimits_{\boldsymbol{\kappa}\in\Omega_{\text{int}}} w_{i,\boldsymbol{\kappa}}\cdot I(\boldsymbol{\kappa}) \\ &= \sum_{j\in N} \underbrace{\sum_{\boldsymbol{\kappa}\in\Omega_j} w_{i,\boldsymbol{\kappa}}\phi(\boldsymbol{\kappa})}_{=a_{i\leftarrow\psi(j)}} + \sum_{S\subseteq N,|S|>1} \underbrace{\sum_{\boldsymbol{\kappa}\in\Omega_S} w_{i,\boldsymbol{\kappa}} I(\boldsymbol{\kappa})}_{=a_{i\leftarrow J(S)}} \end{aligned} \quad (12)$$

where $\Omega_{\text{ind}} = \cup_{j\in N}\Omega_j$ is the set of degree vectors $\boldsymbol{\kappa}$ that corresponds to all Taylor indpendent effects and $\Omega_{\text{int}} = \cup_{S\subseteq N,|S|>1}\Omega_S$ is the set of degree vectors $\boldsymbol{\kappa}$ corresponding to all Taylor interaction effects. In addition, $w_{i,\boldsymbol{\kappa}}$ denotes the ratio of a specific Taylor independent effect $\phi(\boldsymbol{\kappa})$ (Taylor interaction effect $I(\boldsymbol{\kappa})$) that is allocated to a_i. By combining (11) and (12), we can obtain the relationship between $w_{i,j}, w_{i,S}$ and $w_{i,\boldsymbol{\kappa}}$ as follows:

$$w_{i,j} = \frac{\sum_{\boldsymbol{\kappa}\in\Omega_j} w_{i,\boldsymbol{\kappa}}\phi(\boldsymbol{\kappa})}{\psi(j)},\ w_{i,S} = \frac{\sum_{\boldsymbol{\kappa}\in\Omega_S} w_{i,\boldsymbol{\kappa}} I(\boldsymbol{\kappa})}{J(S)} \quad (13)$$

Essential difference among attribution methods. Based on the unified paradigm in (11) and (12), we discover that the essential difference between different attribution methods is that each attribution method actually uses a different ratio $w_{i,j}$, $w_{i,S}$, and $w_{i,\boldsymbol{\kappa}}$ to re-allocate different effects.

Moreover, under the unified paradigm, we discover that not all attribution methods allocate a *reasonable* ratio of each effect to the attribution score a_i. For example, we find that some attribution methods may allocate part of the generic interaction effect $J(S)$ to the variable i that is not involved in the interaction (*i.e.*, $i \notin S$). In addition, some attribution methods may fail to completely allocate all numerical values of the generic interaction effect $J(S)$ to input variables, *e.g.*, $\sum_{i\in N} a_{i\leftarrow J(S)} < J(S)$. Therefore, in Section IV, we propose three principles to examine whether an attribution method reasonably allocates independent effects and interaction effects, to evaluate the faithfulness of attribution methods.

D. Unifying Fourteen Attribution Methods With Interaction Effects and Independent Effects

In this subsection, we reformulate fourteen existing attribution methods into the unified paradigm of allocating Taylor independent effects and interaction effects in (12) one by one. We have summarized all reformulations in Table IV.

Gradient × Input. Gradient × Input [33] estimates the attribution by roughly considering a complex DNN f as a linear model, *i.e.*, $f(\boldsymbol{x})\underline{\text{explained}} f(\boldsymbol{0}) + \sum_i \frac{\partial f(\boldsymbol{x})}{\partial x_i} x_i$. Here, $\frac{\partial f(\boldsymbol{x})}{\partial x_i}$ denotes the gradient of the output *w.r.t.* the input variable i. Thus, Gradient × Input considers that the product of the gradient and input reflects the attribution of the variable i.

$$a_i = \frac{\partial f(\boldsymbol{x})}{\partial x_i} x_i. \quad (14)$$

Theorem 2. (Proof in Appendix B, available online) In the Gradient × Input method, the attribution a_i can be reformulated as

$$a_i = \phi(\boldsymbol{\kappa}) = \frac{\partial f(\boldsymbol{x})}{\partial x_i} x_i. \quad (15)$$

where $\boldsymbol{\kappa} = [\kappa_1, \ldots, \kappa_n]$ is a one-hot degree vector with $\kappa_i = 1$ and $\forall j \neq i, \kappa_j = 0$.

Theorem 2 shows that Gradient × Input follows the paradigm of allocating Taylor interaction effects in (12). Specifically, this method allocates only a specific Taylor independent effect $\phi(\boldsymbol{\kappa})$ of the variable i to the attribution of i.

Occlusion-1. To compute the attribution of the input variable i, Occlusion-1 [40] occludes the variable i by the baseline value b_i and obtains an occluded input $\boldsymbol{x}|_{x_i=b_i}$. Then, Occlusion-1 considers that the output change between the original input $\boldsymbol{x}$ and the occluded input $\boldsymbol{x}|_{x_i=b_i}$ reflects the attribution of the

TABLE IV
FOURTEEN ATTRIBUTION METHODS CAN BE UNIFIED INTO THE SAME PARADIGM OF ALLOCATING TAYLOR INTERACTION EFFECTS

Attribution methods	**Unified paradigm of allocating Taylor interaction effects**
Grad×Input [33]	$a_i = \phi(\boldsymbol{\kappa}),\ \boldsymbol{\kappa} = [0, \cdots, \kappa_i = 1, \cdots, 0]$
Occlusion-1 [40]	$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa})$
Occlusion-patch [40]	$a_i = \sum_{k\in S_j} \sum_{\boldsymbol{\kappa}\in\Omega_k} \phi(\boldsymbol{\kappa}) + \sum_{S\cap S_j\neq\emptyset, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa})$
Prediction Diff [41]	$a_i = \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b})}[\sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}\lvert\boldsymbol{b}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa}\lvert\boldsymbol{b})]$
Grad-CAM [31]	$\tilde{a}_i = \phi(\boldsymbol{\kappa}),\ \boldsymbol{\kappa} = [0, \cdots, \kappa_i = 1, \cdots, 0]$
Integrated Grads [37]	$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \frac{\kappa_i}{\sum_i \kappa_i} \cdot I(\boldsymbol{\kappa})$
Expected Grads [13]	$a_i = \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b})}[\sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}\lvert\boldsymbol{b}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \frac{\kappa_i}{\sum_i \kappa_i} I(\boldsymbol{\kappa}\lvert\boldsymbol{b})]$
Shapley value [22]	$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} 1/\lvert S\rvert \cdot I(\boldsymbol{\kappa})$
LRP-ϵ [3]	$a_i = \phi(\boldsymbol{\kappa}),\ \boldsymbol{\kappa} = [0, \cdots, \kappa_i = 1, \cdots, 0]$
LRP-$\alpha\beta$ [3]	$a_i = \begin{cases} \alpha[\sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} c_i I(\boldsymbol{\kappa}) + \sum_{S\subseteq N^-} \sum_{\boldsymbol{\kappa}\in\Omega_S} d_i I(\boldsymbol{\kappa})], & i \in N^+ \\ \beta[\sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \tilde{c}_i I(\boldsymbol{\kappa}) + \sum_{S\subseteq N^+} \sum_{\boldsymbol{\kappa}\in\Omega_S} \tilde{d}_i I(\boldsymbol{\kappa})], & i \in N^- \end{cases}$
Deep Taylor [23]	$a_i = \begin{cases} \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} c_i I(\boldsymbol{\kappa}) + \sum_{S\subseteq N^-} \sum_{\boldsymbol{\kappa}\in\Omega_S} d_i I(\boldsymbol{\kappa}), & i \in N^+ \\ 0, & i \in N^- \end{cases}$
DeepLIFT Rescale [32]	$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \frac{\kappa_i}{\sum_i \kappa_i} \cdot I(\boldsymbol{\kappa})$
DeepLIFT Reveal [32]	$a_i = \begin{cases} \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{S\subseteq N^+, i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} c_i I(\boldsymbol{\kappa}) \\ + \sum_{S\cap N^+\neq\emptyset, S\cap N^-\neq\emptyset, i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \frac{1}{2} c_i I(\boldsymbol{\kappa})], & i \in N^+ \\ \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{S\subseteq N^-, i\in S} \sum_{\boldsymbol{\kappa}\in\Omega_S, \lvert S\rvert>1} \tilde{c}_i I(\boldsymbol{\kappa}) \\ + \sum_{S\cap N^+\neq\emptyset, S\cap N^-\neq\emptyset, i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \frac{1}{2} \tilde{c}_i I(\boldsymbol{\kappa})], & i \in N^- \end{cases}$
DeepShap [22]	$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, \lvert S\rvert>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} 1/\lvert S\rvert \cdot I(\boldsymbol{\kappa})$

variable i.

$$a_i = f(\boldsymbol{x}) - f(\boldsymbol{x}|_{x_i=b_i}). \tag{16}$$

where $\forall i, b_i = b$ and b is a constant scalar.

Theorem 3. (Proof in Appendix B, available online) In the Occlusion-1 method, the attribution a_i can be reformulated as

$$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa}) \tag{17}$$

Theorem 3 shows that Occlusion-1 follows the paradigm of allocating Taylor interaction effects in (12). This method allocates the generic independent effect $\psi(i) = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa})$ of the variable i to its attribution a_i. In addition, this method allocates each entire generic interaction effect $J(S) = \sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa})$, which involves the variable i ($i \in S$), to the attribution a_i. In other words, the Occlusion-1 method repeatedly allocates the generic interaction effect $J(S)$ to each variable in S.

Occlusion-patch. Occlusion-patch [40] first divides an image into m patches, *i.e.*, $M = \{S_1, \ldots, S_m\}$. To compute attributions of pixels in each patch S_j, Occlusion-patch occludes all pixels in the patch by the baseline value $\boldsymbol{b}$ and obtains an occluded input $\boldsymbol{x}|_{\boldsymbol{x}_{S_j}=\boldsymbol{b}}$. Then, Occlusion-patch considers that the output change between the original input and the occluded input reflects attributions of pixels in S_j.

$$a_i = f(\boldsymbol{x}) - f(\boldsymbol{x}|_{\boldsymbol{x}_{S_j}=\boldsymbol{b}}),\ \forall i \in S_j \tag{18}$$

Theorem 4. (Proof in Appendix B, available online) In the Occlusion-patch method, the attribution of the pixel i in the patch S_j ($i \in S_j$) can be reformulated as

$$a_i = \sum_{m\in S_j} \sum_{\boldsymbol{\kappa}\in\Omega_m} \phi(\boldsymbol{\kappa}) + \sum_{S\cap S_j\neq\emptyset, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa}) \tag{19}$$

Theorem 4 shows that Occlusion-patch follows the paradigm of allocating Taylor interaction effects in (12). For the pixel $i \in S_j$, this method allocates generic independent effects of all pixels in S_j to the attribution a_i, *i.e.*, allocating $\sum_{m\in S_j} \sum_{\boldsymbol{\kappa}\in\Omega_m} \phi(\boldsymbol{\kappa})$ to a_i. Besides, this method allocates all generic interaction effects $J(S) = \sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa})$, which involve some pixels in S_j ($S \cap S_j \neq \emptyset$), to the attribution a_i. Hence, the Occlusion-patch method may mistakenly assign the generic interaction effect $J(S)$, which does not involve the variable i ($i \notin S$), to the variable i's attribution.

Prediction Difference. Prediction Difference [41] is an extension of the Occlusion-1 method [40]. Unlike the Occlusion-1 method simply using a single baseline value to occlude x_i, the Prediction Difference method samples multiple baseline points from a distribution $p(b_i)$. For example, the distribution can be set as the conditional distribution of x_i given other variables, $p(b_i) = p(x_i|x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. However, this method assumes $b_1 = b_2 = \cdots = b_n = b$. Then, the attribution a_i is computed by averaging attributions generated

by the Occlusion-1 method over different baseline points.

$$a_i = \mathbb{E}_{b_i \sim p(b_i)}[f(\boldsymbol{x}) - f(\boldsymbol{x}|_{x_i=b_i})]. \tag{20}$$

Theorem 5. (Proof in Appendix B, available online) In the Prediction Difference method, the attribution a_i is reformulated as

$$a_i = \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b})}\left[\sum_{\boldsymbol{\kappa}\in\Omega_i}\phi(\boldsymbol{\kappa}|\boldsymbol{b}) + \sum_{i\in S,|S|>1}\sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa}|\boldsymbol{b})\right], \tag{21}$$

where $\boldsymbol{b} = b\cdot\mathbf{1}$.

Theorem 5 shows that Prediction Difference follows the paradigm of allocating Taylor interaction effects in (12). This method obtains the attribution a_i by adopting the same allocation strategy as the Occlusion-1 method in (17), and then averages the attributions a_i over different baseline points to obtain the final attribution.

Grad-CAM. Grad-CAM [31] estimates the attribution of neural activations at each location (i, j) in a convolutional layer, as follows.

$$a_{ij} = \text{ReLU}\left(\sum\nolimits_{k=1}^{K}\alpha_k A_{ij}^k\right),$$
$$\text{where } \alpha_k = \frac{1}{W\times H}\sum\nolimits_{i=1}^{W}\sum\nolimits_{j=1}^{H}\frac{\partial y}{\partial A_{ij}^k}. \tag{22}$$

where A_{ij}^k denotes the neuron activation at the (i, j) location in the feature map A^k of the k-th channel. Please see Appendix B for detailed introduction, available online. To simplify the analysis, we just explain the attribution $\tilde{a}_{ij}$ before the ReLU operation in (22), subject to $a_{ij} = ReLU(\tilde{a}_{ij})$.

$$\tilde{a}_{ij} = \sum\nolimits_{k=1}^{K}\alpha_k A_{ij}^k. \tag{23}$$

Theorem 6. (Proof in Appendix B, available online) In the Grad-CAM method, the attribution of each neuron is reformulated as:

$$\tilde{a}_{ij}^k = \phi(\boldsymbol{\kappa}) = \frac{\partial g(F)}{\partial A_{ij}^k}A_{ij}^k. \tag{24}$$

Here, $g(F)$ is the explanatory model of the DNN in the Grad-CAM method, which has been proved in [31]. In addition, $\boldsymbol{\kappa} = [\kappa_1, \ldots, \kappa_n]$ is a one-hot degree vector with $\kappa_i = 1$ and $\forall j \neq i, \kappa_j = 0$.

Theorem 6 shows that Grad-CAM follows the paradigm of allocating Taylor interaction effects in (12). It only allocates a specific Taylor independent effect $\phi(\boldsymbol{\kappa})$ of the neuron A_{ij}^k to its attribution.

By comparing Theorem 2 and Theorem 6, we find that Grad-CAM and Gradient $\times$ Input actually share similar mechanisms of allocating Taylor interaction effects. The main difference between the two methods is that Grad-CAM explains the attribution of features in the convolutional layer, whereas Gradient $\times$ Input explains the attribution of variables in the input layer.

Integrated Gradients. The Integrated Gradients method [37] estimates the attribution a_i as follows.

$$a_i = (x_i - b_i)\cdot\int_{\alpha=0}^{1}\left.\frac{\partial f(c)}{\partial c_i}\right|_{\boldsymbol{c}=(\boldsymbol{b}+\alpha(\boldsymbol{x}-\boldsymbol{b}))} d\alpha \tag{25}$$

This method estimates the attribution as the element-wise product of the *average* gradient and input, where gradients are averaged over numerous points along a linear path from the baseline $\boldsymbol{b}$ to the input $\boldsymbol{x}$.

Theorem 7. (Proof in Appendix B, available online) In the Integrated Gradients method, the attribution a_i is reformulated as

$$a_i = \sum\nolimits_{\boldsymbol{\kappa}\in\Omega_i}\phi(\boldsymbol{\kappa}) + \sum\nolimits_{i\in S,|S|>1}\sum\nolimits_{\boldsymbol{\kappa}\in\Omega_S}\frac{\kappa_i}{\sum_{i'}\kappa_{i'}}I(\boldsymbol{\kappa}) \tag{26}$$

Theorem 7 shows that Integrated Gradients follows the paradigm of allocating Taylor interaction effects in (12). This method allocates the variable i's generic independent effect $\psi(i) = \sum_{\boldsymbol{\kappa}\in\Omega_i}\phi(\boldsymbol{\kappa})$ to the attribution a_i. In addition, this method allocates a specific ratio of each Taylor interaction effect $I(\boldsymbol{\kappa})$ ($\boldsymbol{\kappa}\in\Omega_S, i\in S$), which involves the variable i, to the attribution a_i. The ratio is proportional to the degree κ_i of the variable i.

Expected Gradients. Expected Gradients [13] is an extension of Integrated Gradients [37]. To estimate the attribution, this method samples baseline points from a prior distribution $p(\boldsymbol{b})$ (*e.g.*, $\boldsymbol{b}\sim N(\boldsymbol{x}, \sigma^2\boldsymbol{I})$), instead of specifying a certain baseline in the Integrated Gradients method. Then, the attribution a_i is computed by integrating attributions estimated by the Integrated Gradients method over different baselines.

$$a_i = \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b})}\cdot\left[(x_i - b_i)\int_{\alpha=0}^{1}\left.\frac{\partial f(c)}{\partial c_i}\right|_{\boldsymbol{c}=(\boldsymbol{b}+\alpha(\boldsymbol{x}-\boldsymbol{b}))} d\alpha\right] \tag{27}$$

Theorem 8. (Proof in Appendix B, available online) In the Expected Gradients method, the attribution a_i can be reformulated as:

$$a_i = \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b})}\left[\sum_{\boldsymbol{\kappa}\in\Omega_i}\phi(\boldsymbol{\kappa}|\boldsymbol{b}) + \sum_{i\in S,|S|>1}\sum_{\boldsymbol{\kappa}\in\Omega_S}\frac{\kappa_i}{\sum_{i'}\kappa_{i'}}I(\boldsymbol{\kappa}|\boldsymbol{b})\right] \tag{28}$$

Theorem 8 shows that Expected Gradients follows the paradigm of allocating Taylor interaction effects in (12). This method obtains the attribution by adopting the same allocation strategy as the Integrated Gradients method in (26), and then averages the attributions over different baselines to obtain final attributions.

Shapley value. The Shapley value method [19], [22] estimates the attribution of each variable as follows.

$$a_i = \sum\nolimits_{S\subseteq N\setminus\{i\}} p(S)\cdot[f(\boldsymbol{x}_{S\cup\{i\}}) - f(\boldsymbol{x}_S)]. \tag{29}$$

where $p(S) = |S|!(n-1-|S|)!/n!$. The Shapley value method formulates the attribution of the variable i as its average marginal contribution $f(\boldsymbol{x}_{S\cup\{i\}}) - f(\boldsymbol{x}_S)$ over different

contextual subsets S. Here, $f(\boldsymbol{x}_S)$ is computed as the network output when variables in $N \setminus S$ are masked and variables in S keep unchanged.

Theorem 9. (Proof in Appendix B, available online) In the Shapley value method, the attribution a_i can be reformulated as

$$a_i = \sum\nolimits_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum\nolimits_{i\in S} \sum\nolimits_{\boldsymbol{\kappa}\in\Omega_S, |S|>1} 1/|S| \cdot I(\boldsymbol{\kappa}) \tag{30}$$

Theorem 9 shows that Shapley value follows the paradigm of allocating Taylor interaction effects in (12). This method allocates the generic independent effect $\psi(i) = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa})$ of the variable i to the attribution a_i. Furthermore, this method allocates each Taylor interaction effect $I(\boldsymbol{\kappa})$ $(\boldsymbol{\kappa} \in \Omega_S, i \in S)$, which involves the variable i, to the attribution a_i. The effect $I(\boldsymbol{\kappa})$ is uniformly allocated to all the $s = |S|$ variables involved in the interaction S, *i.e.*, each input variable receives $1/|S| \cdot I(\boldsymbol{\kappa})$.

Back-propagation attribution methods. Among various attribution methods, a typical type of method is designed to estimate the attribution of each feature dimension at an intermediate layer, and then back-propagate these attributions to previous layers until the input layer. That is, $\boldsymbol{a}^{(L)} \to \boldsymbol{a}^{(L-1)} \to \cdots \to \boldsymbol{a}^{(1)} \to \boldsymbol{a}^{(0)}$, where $\boldsymbol{a}^{(l)} \in \mathbb{R}^{n_l}$ denotes attributions of all feature dimensions in the l-th layer. In particular, $\boldsymbol{a}^{(0)} \in \mathbb{R}^n$ corresponds to attributions in the input layer. This type of method is known as *back-propagation attribution methods*, including LRP-ϵ [3], LRP-$\alpha\beta$ [3], Deep Taylor [23], DeepLIFT Rescale [32], Deep SHAP [22], DeepLIFT RevealCancel [32], and so on.

The *essential difference* between various back-propagation methods is that they employ different recursive rules for back-propagating attributions through two adjacent layers, *i.e.*, $\boldsymbol{a}^{(l)} \to \boldsymbol{a}^{(l-1)}$. In particular, these methods usually simplify various layer-wise operations in different DNNs as a typical module $\boldsymbol{x}^{(l)} = \sigma(W\boldsymbol{x}^{(l-1)} + \boldsymbol{s})$, which serve as a representative. Here, $\boldsymbol{x}^{(l)}$ denotes the feature in the l-th layer. W and $\boldsymbol{s}$ denote the weight and the additive bias, respectively. σ is the activation function.

LRP-ϵ. LRP-ϵ [3] is a typical back-propagation attribution method. For the typical module $\boldsymbol{x}^{(l)} = \sigma(W\boldsymbol{x}^{(l-1)} + \boldsymbol{s})$, LRP-$\epsilon$ propagates the following attribution $a^{(l)}_{i\leftarrow j}$ from the neuron j in the l-th layer to the neuron i in the $(l-1)$-th layer.

$$a^{(l)}_{i\leftarrow j} = \begin{cases} \frac{z_{ij}}{(\sum_{i'} z_{i'j} + s_j) + \epsilon} \cdot a^{(l)}_j, & \sum_{i'} z_{i'j} + s_j \geq 0 \\ \frac{z_{ij}}{(\sum_{i'} z_{i'j} + s_j) - \epsilon} \cdot a^{(l)}_j, & \sum_{i'} z_{i'j} + s_j < 0 \end{cases} \tag{31}$$

where $z_{ij} = W_{ij}x^{(l-1)}_i$. Since $x^{(l)}_j = \sigma(\sum_{i'} z_{i'j} + s_j)$, LRP-$\epsilon$ considers that z_{ij} can reflect the contribution of $x^{(l-1)}_i$ to $x^{(l)}_j$, to some extent. To avoid dividing 0, LRP-ϵ introduces a small quantity $\epsilon > 0$ in the denominator. Then, LRP-ϵ formulates the attribution $a^{(l-1)}_i$ as the sum of these propagated values from all feature dimensions in the l-th layer, *i.e.*, $a^{(l-1)}_i = \sum_j a^{(l)}_{i\leftarrow j}$.

Theorem 10. (Proof in Appendix B, available online) When ReLU is used as the activation function, the attribution a_i estimated by the LRP-ϵ method can be reformulated as

$$a_i = \phi(\boldsymbol{\kappa}) = \frac{\partial f(\boldsymbol{x})}{\partial x_i} x_i \tag{32}$$

where $\boldsymbol{\kappa} = [\kappa_1, \ldots, \kappa_n]$ is a one-hot degree vector with $\kappa_i = 1$ and $\forall j \neq i, \kappa_j = 0$.

Theorem 10 shows that LRP-ϵ follows the paradigm of allocating Taylor interaction effects in (12). Specifically, this method allocates only a specific Taylor independent effect $\phi(\boldsymbol{\kappa})$ of the input variable i to the attribution of the input variable i.

By comparing Theorems 2 and 10, it is easy to find that LRP-ϵ and Gradient $\times$ Input are essentially the same, because they allocate Taylor interaction effects in the same way when ReLU is adopted as the activation function. Furthermore, Fig. 3 also verifies that the two methods produce the same attribution results.

LRP-$\alpha\beta$. LRP-$\alpha\beta$ [3] is also a typical back-propagation attribution method. It slightly modifies the recursive propagation rule of LRP-ϵ as follows,

$$a^{(l)}_{i\leftarrow j} = \begin{cases} \frac{\alpha \cdot z_{ij}}{\sum_{i'\in N^+} z_{i'j}} \cdot a^{(l)}_j, & i \in N^+ \\ \frac{\beta \cdot z_{ij}}{\sum_{i'\in N^-} z_{i'j}} \cdot a^{(l)}_j, & i \in N^- \end{cases} \tag{33}$$

Unlike LRP-ϵ, LRP-$\alpha\beta$ divides all neurons in the $(l-1)$-th layer into two groups, *i.e.*, $N^+ = \{i | z_{ij} > 0\}$ with positive contributions and $N^- = \{i | z_{ij} \leq 0\}$ with negative contributions. Then, LRP-$\alpha\beta$ computes the attribution in the N^+ and N^- group using the weights α and β, respecitively. The final attribution is computed as $a^{(l-1)}_i = \sum_j a^{(l)}_{i\leftarrow j}$.

Theorem 11. (Proof in Appendix B, available online) Let us consider $x^{(l)}_j$ as the target output and $\boldsymbol{x}^{(l-1)} = [x^{(l-1)}_1, \ldots, x^{(l-1)}_{n_l}]$ as input variables, to analyze the layer-wise propagation of attributions. Then, in the LRP-$\alpha\beta$ method, the attribution a_i in N^+ estimated can be reformulated as

$$a_i = \alpha \left[\sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} c_i I(\boldsymbol{\kappa}) + \sum_{S\subseteq N^-} \sum_{\boldsymbol{\kappa}\in\Omega_S} d_i I(\boldsymbol{\kappa}) \right] \tag{34}$$

where $c_i = \kappa_i / \sum_{i'\in N^+} \kappa_{i'}$ and $d_i = z_{ij} / \sum_{i'\in N^+} z_{i'j}$. For variables in N^-,

$$a_i = \beta \left[\sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \tilde{c}_i I(\boldsymbol{\kappa}) + \sum_{S\subseteq N^+} \sum_{\boldsymbol{\kappa}\in\Omega_S} \tilde{d}_i I(\boldsymbol{\kappa}) \right] \tag{35}$$

where $\tilde{c}_i = \kappa_i / \sum_{i'\in N^-} \kappa_{i'}$ and $\tilde{d}_i = z_{ij} / \sum_{i'\in N^-} z_{i'j}$.

Theorem 11 shows that LRP-$\alpha\beta$ follows the paradigm of allocating Taylor interaction effects in (12). As (34) shows, for the variable $i \in N^+$, this method allocates part of i's Taylor independent effect $\phi(\boldsymbol{\kappa})$ $(\boldsymbol{\kappa} \in \Omega_i)$ to the attribution a_i. Besides, this method allocates part of the Taylor interaction effect $I(\boldsymbol{\kappa})$ $(\boldsymbol{\kappa} \in \Omega_S, i \in S)$, which involves the variable i, to the attribution a_i. However, this method mistakenly allocates part of the Taylor interaction effect $I(\boldsymbol{\kappa})$ $(\boldsymbol{\kappa} \in \Omega_S, S \subseteq N^-)$ between some variables in N^- to the attribution of the variable $i \in N^+$. The allocation strategy of variables in N^- is analogous.

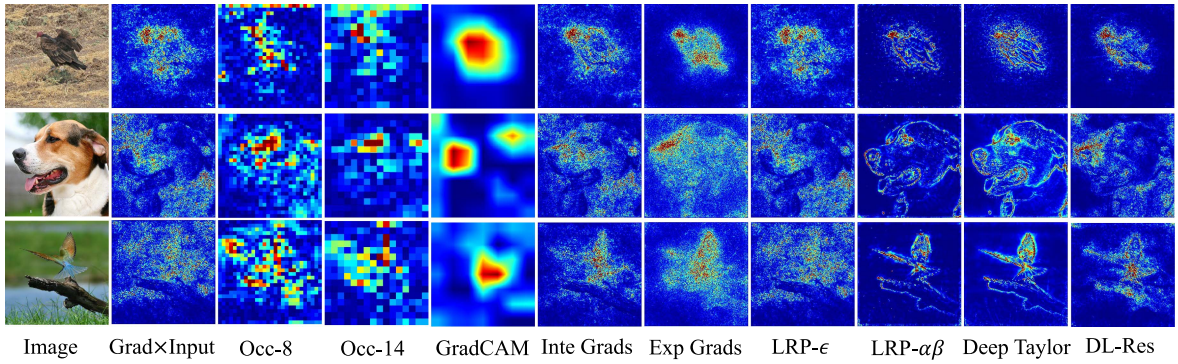


Fig. 3. Attribution maps generated by different attribution methods.

Deep Taylor. Deep Taylor [23] is a typical back-propagation attribution method. For the typical module $\boldsymbol{x}^{(l)} = \sigma(W\boldsymbol{x}^{(l-1)} + \boldsymbol{s})$, it designs the recursive back-propagation rule as follows.

$$a_{i\leftarrow j}^{(l)} = \begin{cases} \frac{z_{ij}}{\sum_{i'\in N^+} z_{i'j}} \cdot a_j^{(l)}, & i \in N^+ \\ 0, & i \in N^- \end{cases} \tag{36}$$

The final attribution is computed as $a_i^{(l-1)} = \sum_j a_{i\leftarrow j}^{(l)}$. In particular, Deep Taylor can be regarded as a special case of LRP-$\alpha\beta$ [3] with $\alpha = 1, \beta = 0$ in (33).

Theorem 12. (Proof in Appendix B, available online) Let us consider $x_j^{(l)}$ as the target output and $\boldsymbol{x}^{(l-1)} = [x_1^{(l-1)}, \ldots, x_{n_l}^{(l-1)}]$ as input variables, to analyze the layer-wise propagation of attributions. Then, the attribution a_i $(i \in N^+)$ in the Deep Taylor method is reformulated as

$$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} c_i I(\boldsymbol{\kappa}) + \sum_{S\subseteq N^-} \sum_{\boldsymbol{\kappa}\in\Omega_S} d_i I(\boldsymbol{\kappa}) \tag{37}$$

where $c_i = \kappa_i / \sum_{i'\in N^+} \kappa_{i'}$, and $d_i = z_{ij} / \sum_{i'\in N^+} z_{i'j}$. *Moreover, for the variable $i \in N^-$, $a_i = 0$.*

Theorem 12 shows that Deep Taylor follows the paradigm of allocating Taylor interaction effects in (12). The weight of allocation is almost the same as the weight of LRP-$\alpha\beta$ in (34) and (35), and differs only by a constant.

DeepLIFT Rescale. DeepLIFT Rescale [32] is also a typical back-propagation attribution method, which recursively propagates the attribution as follows.

$$a_{i\leftarrow j}^{(l)} = \frac{\Delta z_{ij}}{\sum_{i'}(\Delta z_{i'j})} \cdot a_j^{(l)} \tag{38}$$

where $\Delta z_{ij} = z_{ij} - \tilde{z}_{ij}$, $z_{ij} = W_{ij}x_i^{(l-1)}$, $\tilde{z}_{ij} = W_{ij}\tilde{x}_i^{(l-1)}$. Here, $\tilde{x}_i^{(l-1)}$ is the selected baseline value to represent the state when $x_i^{(l-1)}$ does not receive any information. Thus, Δz_{ij} reflects the contribution of $x_i^{(l-1)}$ on changing $x_j^{(l)}$ from the state of the baseline value to the current activation value. The final attribution is computed as $a_i^{(l-1)} = \sum_j a_{i\leftarrow j}^{(l)}$.

Theorem 13. (Proof in Appendix B, available online) Let us consider $x_j^{(l)}$ as the target output and $\boldsymbol{x}^{(l-1)} = [x_1^{(l-1)}, \ldots, x_{n_l}^{(l-1)}]$ as input variables, to analyze the layer-wise propagation of attributions. Then, the attribution a_i estimated by the DeepLIFT Rescale method can be reformulated as

$$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \frac{\kappa_i}{\sum_{i'}\kappa_{i'}} I(\boldsymbol{\kappa}) \tag{39}$$

Theorem 13 shows that DeepLIFT Rescale follows the paradigm of allocating Taylor interaction effects in (12). This method allocates the variable i's generic independent effect $\psi(i) = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa})$ to the attribution a_i. Besides, it allocates a specific ratio of each Taylor interaction effect $I(\boldsymbol{\kappa})$ $(\boldsymbol{\kappa} \in \Omega_S, i \in S)$, which involves the variable i, to the attribution a_i. The ratio is proportional to the degree κ_i.

DeepLIFT RevealCancel. DeepLIFT RevealCancel [32] is a typical back-propagation attribution method, which modifies the recursive back-propagation rule of the DeepLIFT Rescale method as follows.

$$a_{i\leftarrow j}^{(l)} = \begin{cases} \frac{\Delta z_{ij}}{\sum_{i'\in N^+} \Delta z_{i'j}} \cdot \frac{\Delta y^+}{\Delta y^+ + \Delta y^-} \cdot a_j^{(l)}, & i \in N^+ \\ \frac{\Delta z_{ij}}{\sum_{i'\in N^-} \Delta z_{i'j}} \cdot \frac{\Delta y^-}{\Delta y^+ + \Delta y^-} \cdot a_j^{(l)}, & i \in N^- \end{cases} \tag{40}$$

Both DeepLIFT Rescale and DeepLIFT RevealCancel use Δz_{ij} (defined in (38)) to represent the contribution of $x_i^{(l-1)}$ on $x_j^{(l)}$. However, the difference is that DeepLIFT RevealCancel divides all contributions into two groups, $N^+ = \{i|\Delta z_{ij} > 0\}$ with positive contributions and $N^- = \{i|\Delta z_{ij} \le 0\}$ with negative contributions. Then, this method computes attributions in N^+ using the weight $\Delta y^+/(\Delta y^+ + \Delta y^-)$ and obtains attributions in N^- using the weight $\Delta y^-/(\Delta y^+ + \Delta y^-)$, respectively. Here, Δy^+ is set to the average marginal contribution of N^+ when all neurons in N^- are present and when all neurons in N^- are absent. Similarly, Δy^- is set to the average marginal contribution of N^- when all neurons in N^+ are present and when all neurons in N^+ are absent. Please see Appendix B, available online for the computation of Δy^+ and Δy^-.

Theorem 14. (Proof in Appendix B, available online) Let us consider $x_j^{(l)}$ as the target output and $\boldsymbol{x}^{(l-1)} = [x_1^{(l-1)}, \ldots, x_{n_l}^{(l-1)}]$ as input variables, to analyze the layer-wise propagation of attributions. Then, in the DeepLIFT RevealCancel method, the attribution $a_i (i \in N^+)$ can be reformulated as

follows.

$$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{S\subseteq N^+, i\in S, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} c_i I(\boldsymbol{\kappa}) + \sum_{S\cap N^+\neq\emptyset, S\cap N^-\neq\emptyset, i\in S, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \frac{1}{2} c_i I(\boldsymbol{\kappa})] \quad (41)$$

The attribution $a_i (i \in N^-)$ *can be reformulated as*

$$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{S\subseteq N^-, i\in S, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \tilde{c}_i I(\boldsymbol{\kappa}) + \sum_{S\cap N^+\neq\emptyset, S\cap N^-\neq\emptyset, i\in S, , |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} \frac{1}{2} \tilde{c}_i I(\boldsymbol{\kappa})] \quad (42)$$

where $c_i = \frac{\kappa_i}{\sum_{i'\in N^+}\kappa_{i'}}$ *and* $\tilde{c}_i = \frac{\kappa_i}{\sum_{i'\in N^-}\kappa_{i'}}$.

Theorem 14 shows that the DeepLIFT RevealCancel method follows the paradigm of allocating Taylor interaction effects in (12). As (41) shows, for the variable $i \in N^+$, this method allocates the variable i's generic independent effect $\psi(i) = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa})$ to the attribution a_i. Besides, this method allocates part of the Taylor interaction effect $I(\boldsymbol{\kappa})$ ($\boldsymbol{\kappa} \in \Omega_S, S \subseteq N^+, i \in S$) between variables in $S \subseteq N^+$, which involves the variable i, to the attribution a_i. Moreover, this method allocates a different ratio of the Taylor interaction effect $I(\boldsymbol{\kappa})$ ($S \cap N^+ \neq \emptyset$, $S \cap N^- \neq \emptyset, i \in S$) between variables in N^+ and variables in N^-, to the attribution a_i. The allocation strategy of variables in N^- is analogous.

Deep SHAP. Deep SHAP [22] is a typical back-propagation method, which combines the Shapley value [19], [22] into the recursive propagation process.

$$a^{(l)}_{i\leftarrow j} = \frac{\phi_i(x^{(l)}_j)}{\sum_{i'}\phi_{i'}(x^{(l)}_j)} \cdot a^{(l)}_j \quad (43)$$

where $\phi_i(x^{(l)}_j)$ denotes the Shapley value of $x^{(l-1)}_i$ *w.r.t.* $x^{(l)}_j$ when we consider $x^{(l)}_j$ as the output and consider features $\boldsymbol{x}^{(l-1)}$ as input variables. Finally, the attribution $a^{(l-1)}_i$ is computed as $a^{(l-1)}_i = \sum_j a^{(l)}_{i\leftarrow j}$.

Theorem 15. (Proof in Appendix B, available online) Let us consider $x^{(l)}_j$ *as the target output and* $\boldsymbol{x}^{(l-1)} = [x^{(l-1)}_1, \ldots, x^{(l-1)}_{n_l}]$ *as input variables, to analyze the layer-wise propagation of attributions. Then, the attribution* a_i *in the Deep SHAP method can be reformulated as follows.*

$$a_i = \sum_{\boldsymbol{\kappa}\in\Omega_i} \phi(\boldsymbol{\kappa}) + \sum_{i\in S, |S|>1} \sum_{\boldsymbol{\kappa}\in\Omega_S} 1/|S| \cdot I(\boldsymbol{\kappa}) \quad (44)$$

Theorem 15 shows that Deep SHAP follows the paradigm of allocating Taylor interaction effects in (12). In each layer of propagation, this method adopts the same allocation strategy as the Shapley value method in (30), to obtain the attribution.

E. Experimental Verification

In this section, we conduct experiments to check the correctness of Theorems 2-15, *i.e.*, whether the theoretically reformulated attributions really reflect true attributions estimated by different methods.

Let us use a specific attribution method to explain a DNN on a given input sample $\boldsymbol{x}$. We use the following metric to measure the average fitting error between the theoretically reformulated attribution values $\boldsymbol{a}(\boldsymbol{x}) \in \mathbb{R}^n$ and the true attribution values $\boldsymbol{a}^*(\boldsymbol{x}) \in \mathbb{R}^n$ estimated by the attribution method.

$$E = \mathbb{E}_{\boldsymbol{x}} \frac{\|\boldsymbol{a}(\boldsymbol{x}) - \boldsymbol{a}^*(\boldsymbol{x})\|_2}{\|\boldsymbol{a}^*(\boldsymbol{x})\|_2} \times 100\%. \quad (45)$$

For example, according to (16) and Theorem 3, $a_i(\boldsymbol{x}) = \sum_{\boldsymbol{\kappa}\in\Omega_i}\phi(\boldsymbol{\kappa}) + \sum_{i\in S}\sum_{\boldsymbol{\kappa}\in\Omega_S} I(\boldsymbol{\kappa})$ and $a^*_i(\boldsymbol{x}) = f(\boldsymbol{x}) - f(\boldsymbol{x}|_{x_i=b_i})$ in the Occlusion-1 method.

Note that it is impossible for us to enumerate all Taylor interaction effects $I(\boldsymbol{\kappa})$ in (1). Thus, it is difficult for us to precisely compute the theoretically reformulated attribution values $\boldsymbol{a}(\boldsymbol{x})$. Instead, given a DNN and an input sample $\boldsymbol{x}$, we compute only the first-order and the second-order Taylor interaction effects $I(\boldsymbol{\kappa})$, which subject to $\boldsymbol{\kappa} \in O = \{\boldsymbol{\kappa} \in \mathbb{N}^n | \kappa_1 + \cdots + \kappa_n = 1 \text{ or } 2\}$. Then, we estimate $\boldsymbol{a}(\boldsymbol{x})$ by ignoring all Taylor interaction effects of greater than two orders for implementation. For example, in the Occlusion-1 method, $a_i(\boldsymbol{x}) \approx \sum_{\boldsymbol{\kappa}\in\Omega_i, \boldsymbol{\kappa}\in O}\phi(\boldsymbol{\kappa}) + \sum_{i\in S}\sum_{\boldsymbol{\kappa}\in\Omega_S, \boldsymbol{\kappa}\in O} I(\boldsymbol{\kappa})$. Note that we do not conduct the Taylor expansion at the input sample $\boldsymbol{x}$ as our preliminary conference version [6]. Instead, we expand at a pre-defined baseline point $\boldsymbol{b}$, which is more standard than the previous version. The baseline $\boldsymbol{b}$ is generated by adding a random Gaussian perturbation on the input sample. Furthermore, we notice that the gating states of ReLU networks do not have continuous gradients, which may introduce a large measurement error. Hence, we train only DNNs with sigmoid activation functions and softplus activation functions (the approximation function of ReLU), rather than DNNs with ReLU activation functions, for testing.

We test $\boldsymbol{a}(\boldsymbol{x})$ and $\boldsymbol{a}^*(\boldsymbol{x})$ on three types of models. The first type of model is the second-order polynomial model, *i.e.*, $f(\boldsymbol{x}) = \sum_{i\in N} c_i x_i + \sum_{i\in N}\sum_{j\in N} c_{ij} x_i x_j$, where c_i and c_{ij} denote model weights. We term this type of model *Polynomial model*. The second and third type of models are the three-layer multi-layer perceptron networks, which apply the sigmoid activation function and softplus activation funtion, respectively. We term them *MLP-Sigmoid* and *MLP-Softplus*, respectively. We train these models on the MNIST dataset [8] and compute the average fitting errors E according to (45).

We evaluate the fitting errors E of seven attribution methods, including the *Gradient* $\times$ *Input*, *Occlusion-1*, *Occlusion-patch*, *Prediction Difference*, *Integrated Gradients*, *Expected Gradients*, and *Shapley value* methods. We do not test back-propagation attribution methods because theoretically reformulation for these methods mainly explain layer-wise propagation rules.

Table V lists average fitting errors E of the seven attribution methods. We find that on the three types of models, fitting errors

TABLE V
AVERAGE FITTING ERRORS BETWEEN THEORETICALLY REFORMULATED ATTRIBUTIONS AND ACTUAL ATTRIBUTION VALUES

Methods	Poly nomial	MLP Sigmoid	MLP Softplus
Grad×Input	0	0	0
Occ-1	0	2.46%	2.64 %
Occ-2×2	0	2.36%	2.67 %
PreDiff	0	2.69%	3.36 %
Inte Grads	0.12%	0.82%	0.93 %
Exp Grads	0.16%	0.90%	1.39 %
Shapley	0	1.18%	1.83 %

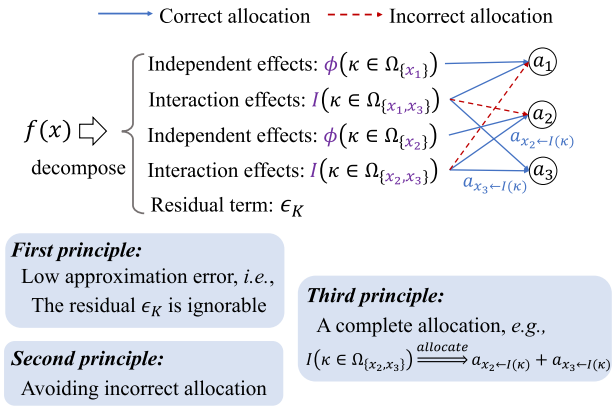


Fig. 4. Illustration of three principles to evaluate the faithfulness of attribution methods.

E of different attribution methods are very small. Theoretically, there should not be any fitting errors when we test on Polynomial models, but tiny errors of the *Integrated Gradients* and *Expected Gradients* methods come from the unavoidable error of their integral computation. The above result indicates that for these attribution methods, the theoretically derived attribution value $\boldsymbol{a}(\boldsymbol{x})$ well fits their actual attribution value $\boldsymbol{a}^*(\boldsymbol{x})$ in real applications.

IV. EVALUATING ATTRIBUTION METHODS

In the last section, we have proven that various attribution scores estimated by fourteen attribution methods can all be reformulated into the *unified paradigm* of allocating Taylor independent effects $\phi(\boldsymbol{\kappa})$ and Taylor interaction effects $I(\boldsymbol{\kappa})$ in (12).

The above finding enables us to directly and fairly evaluate different attribution methods under the same unified paradigm. Therefore, in this section, we propose three principles to evaluate the faithfulness of the fourteen attribution methods.

A. Principles for a Faithful Attribution Method

As shown in Fig. 4, our unified paradigm indicates that each attribution method can be considered as a two-step flowchart, (i) *Taylor expansion*: which first represents the DNN as a Taylor expansion model, and *re-allocation*: which then accordingly re-allocates Taylor independent effects $\phi(\boldsymbol{\kappa})$ and Taylor interaction effects $I(\boldsymbol{\kappa})$ to compute the attribution a_i.

To this end, we find that the faithfulness of an attribution method depends on two key factors:

(i) whether the residual term ϵ_K is small enough in the Taylor expansion of the DNN; (ii) whether the Taylor independent effects and the Taylor interaction effects are allocated to input variables in a reasonable manner.

Accordingly, we propose three principles that faithful attributions are supposed to follow.

• *First principle: low approximation error.* In the Taylor expansion, faithful attribution explanations are expected to cover almost all expansion terms of the DNN, and only leave an ignorable residual term ϵ_K not been explained.

• *Second principle: avoiding unrelated allocation.*

(i) In the re-allocation, *each Taylor independent effect $\phi(\boldsymbol{\kappa})(\boldsymbol{\kappa} \in \Omega_i)$ of the variable i, is supposed to be allocated only to i's attribution*. More precisely, we can decompose a term $a_{i \leftarrow \phi(\boldsymbol{\kappa})} = w_{i,\boldsymbol{\kappa}} \phi(\boldsymbol{\kappa})$ from the attribution a_i according to (12), to represent the numerical effect assigned from the Taylor independent effect $\phi(\boldsymbol{\kappa})$ to the variable i. Then, we should avoid allocating i's independent effect to other unrelated variables $j \neq i$.

$$\forall \boldsymbol{\kappa} \in \Omega_i,\ \forall j \neq i,\ \ a_{j \leftarrow \phi(\boldsymbol{\kappa})} = 0. \tag{46}$$

(ii) In the re-allocation, *each Taylor interaction effect $I(\boldsymbol{\kappa})$ $(\boldsymbol{\kappa} \in \Omega_S)$ between variables in S, is supposed to exclusively be allocated to variables in S, without being allocated to any other unrelated variables $j \notin S$.*

$$\forall \boldsymbol{\kappa} \in \Omega_S,\ \forall j \notin S,\ \ a_{j \leftarrow I(\boldsymbol{\kappa})} = 0. \tag{47}$$

where $a_{j \leftarrow I(\boldsymbol{\kappa})} = w_{j,\boldsymbol{\kappa}} I(\boldsymbol{\kappa})$ denotes the numerical effect assigned from the Taylor interaction effect $I(\boldsymbol{\kappa})$ to the input variable j, according to (12).

• *Third principle: complete allocation.* In the re-allocation, *each Taylor independent effect $\phi(\boldsymbol{\kappa})$* with $\boldsymbol{\kappa} \in \Omega_{\text{ind}}$ (i.e., $\forall i \in N, \forall \boldsymbol{\kappa} \in \Omega_i$) *is supposed to* completely*be allocated to input variables.* In other words, if we accumulate all numerical effects allocated from $\phi(\boldsymbol{\kappa})$ to different variables, we can obtain the exact value of $\phi(\boldsymbol{\kappa})$.

$$\forall \boldsymbol{\kappa} \in \Omega_{\text{ind}},\ \ \sum_{j \in N} a_{j \leftarrow \phi(\boldsymbol{\kappa})} = \phi(\boldsymbol{\kappa}). \tag{48}$$

Similarly, *each Taylor interaction effect $I(\boldsymbol{\kappa})$* with $\boldsymbol{\kappa} \in \Omega_{\text{int}}$ (i.e., $\forall S \subseteq N, |S| > 1, \forall \boldsymbol{\kappa} \in \Omega_S$) *is supposed to* completely*be allocated to different variables.*

$$\forall \boldsymbol{\kappa} \in \Omega_{\text{int}},\ \ \sum_{j \in N} a_{j \leftarrow I(\boldsymbol{\kappa})} = I(\boldsymbol{\kappa}). \tag{49}$$

B. Evaluating Attribution Methods

In this subsection, we use the proposed principles to evaluate the above fourteen attribution methods, which is summarized in Table VI.

TABLE VI
SUMMARY OF PRINCIPLES FOLLOWED BY EACH ATTRIBUTION METHOD

Attribution methods	low approximation error	no unrelated allocation	complete allocation	Attribution methods	low approximation error	no unrelated allocation	complete allocation
Grad×Input	×	✓	✓	Shapley	✓	✓	✓
Occ-1	✓	✓	×	LRP-ϵ	×	✓	✓
Occ-patch	✓	×	×	LRP-$\alpha\beta$	✓	×	✓
PreDiff	✓	✓	×	Deep Taylor	✓	×	✓
Grad-CAM	×	✓	✓	DeepLIFT Res	✓	✓	✓
InteGrads	✓	✓	✓	DeepLIFT Rev	✓	✓	✓
ExpGrads	✓	✓	✓	DeepShap	✓	✓	✓

• Gradient × Input, LRP-ϵ, and Grad-CAM do not satisfy the *low-approximation-error* principle. According to Theorems 2, 6, and 10, these methods consider only the first-order Taylor expansion terms of the DNN to compute attributions, and ignore expansion terms of higher orders.

• Deep Taylor, LRP-$\alpha\beta$, and Occlusion-patch all violate the principle of *avoiding unrelated allocation*. According to Theorems 11 and 12, Deep Taylor and LRP-$\alpha\beta$ mistakenly allocate the Taylor interaction effects $I(\boldsymbol{\kappa})$ between variables in N^- ($\boldsymbol{\kappa} \in \Omega_S, S \subseteq N^-$), to variables $i \in N^+$ that are unrelated to this interaction. Besides, according to Theorem 4, Occlusion-patch mistakenly allocates the Taylor interaction effect, which does not involve the variable i, to the unrelated variable i.

• Occlusion-1, Occlusion-patch, and Prediction Difference all violate the *complete-allocation* principle. According to Theorems 3, 4, and 5, the three methods *repeatedly* allocate the entire Taylor interaction effect $I(\boldsymbol{\kappa})$ between variables in S ($\boldsymbol{\kappa} \in \Omega_S$), to each variable in S. That is, $\forall i \in S,\ a_{i\leftarrow I(\boldsymbol{\kappa})} = I(\boldsymbol{\kappa})$. In this way, the sum of numerical effects allocated from $I\boldsymbol{\kappa})$ to different variables is $|S|$ times as much as $I(\boldsymbol{\kappa})$, *i.e.*, $\sum_{i\in N} a_{i\leftarrow I(\boldsymbol{\kappa})} = |S| \cdot I(\boldsymbol{\kappa}) \neq I(\boldsymbol{\kappa})$, which violates the *complete-allocation* principle in (49).

• According to Theorems 7, 8, 9, 13 and 14, the Integrated Gradients, Expected Gradients, Shapley value, Deep Shap, DeepLIFT Rescale, and DeepLIFT RevealCancel methods satisfy all principles.

For clarity, let us consider a specific Taylor interaction effect $I(\boldsymbol{\kappa}) = x_1^2 x_2 x_3^2$ in a polynomial function as a toy example to illustrate the effect allocation, where the degrees $\kappa_1 = 2, \kappa_2 = 1, \kappa_3 = 2$. Here, $I(\boldsymbol{\kappa})$ quantifies the interaction effect between variables in the set $S = \{x_1, x_2, x_3\}$. In this example, the above methods all allocate a specific weight of the interaction effect $I(\boldsymbol{\kappa})$ to the variable i, *i.e.*, $a_{i\leftarrow I(\boldsymbol{\kappa})} = w \cdot I(\boldsymbol{\kappa})$. Their difference mainly lies in the weights w of allocation. For example, the weight w in both Integrated Gradients and Expected Gradients methods are the relative degree of the variable i, *i.e.*, $w = \kappa_i / \sum_{i'} \kappa_{i'}$. In this way, the two attribution methods allocate $a_{1\leftarrow I(\boldsymbol{\kappa})} = 2/5 \cdot I(\boldsymbol{\kappa})$, $a_{2\leftarrow I(\boldsymbol{\kappa})} = 1/5 \cdot I(\boldsymbol{\kappa})$, $a_{3\leftarrow I(\boldsymbol{\kappa})} = 2/5 \cdot I(\boldsymbol{\kappa})$ to the variables x_1, x_2, x_3, respectively. Different attributions generated by the two methods are caused by the fact that they use different baseline points. In addition, the weight of allocation in the Shapley value method is $w = 1/|S|$, which means $a_{1\leftarrow I(\boldsymbol{\kappa})} = a_{2\leftarrow I(\boldsymbol{\kappa})} = a_{3\leftarrow I(\boldsymbol{\kappa})} = 1/3 \cdot I(\boldsymbol{\kappa})$.

The suitability of an attribution method depends on the specific task for the DNN. For example, in the image classification task, the attribution of each pixel generated by Integrated Gradients may be biased. This is because according to Theorems 7, Integrated Gradients usually allocates a greater Taylor interaction effect to the pixel with a more significant pixel value (*e.g.*, white pixels). In comparison, the Shapley value, which uniformly allocates Taylor interaction effects to different pixels involved in the interaction, may be more suitable for the image classification task.

Our principles provide a new perspective, allowing for a fair evaluation of the faithfulness of attribution methods within the same theoretical system. Note that this does not imply that attribution methods satisfying these principles are ideal attributions. Many other perspectives have been proposed to evaluate attribution methods [1], [2], [15], [21], [39].

C. Connections to the Infidelity metric [39]

We find that some attribution methods, which satisfy the proposed three principles, also have high rankings when they are evaluated by the infidelity metric [39].

Specifically, for one attribution method, the infidelity metric is proposed to evaluate whether attribution explanations $\boldsymbol{a}$ generated by the attribution method can *well reflect the output change under input perturbations*. Specifically, given a DNN f, an input sample $\boldsymbol{x} \in \mathbb{R}^n$ and estimated attribution scores $\boldsymbol{a} \in \mathbb{R}^n$, the infidelity metric is defined as

$$\text{INFD}(\boldsymbol{a}, f, \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{p}_{\boldsymbol{x}}}[\boldsymbol{p}_{\boldsymbol{x}}^T \boldsymbol{a} - (f(\boldsymbol{x}) - f(\boldsymbol{x} - \boldsymbol{p}_{\boldsymbol{x}}))]^2 \quad (50)$$

where $\boldsymbol{p}_{\boldsymbol{x}} \in \mathbb{R}^n$ denotes the perturbation added on the sample $\boldsymbol{x}$. Thus, the infidelity metric quantifies the average error of using attributions to predict the output change $f(\boldsymbol{x}) - f(\boldsymbol{x} - \boldsymbol{p}_{\boldsymbol{x}})$ *w.r.t.* input perturbations. One attribution method with low infidelity is indicative of superior performance. In implementation, we adopt the square removal perturbation in [39] for evaluation. We evaluate the infidelity on a three-layer MLP network with sigmoid and softplus activation functions. We also evaluate the infidelity on a three-layer CNN network with sigmoid and softplus activation functions. These networks are trained on the MNIST dataset.

Fig. 5(a) shows the relationship between the proposed three principles and the infidelity metric. The x-axis denotes whether one attribution method satisfies the proposed principles, and

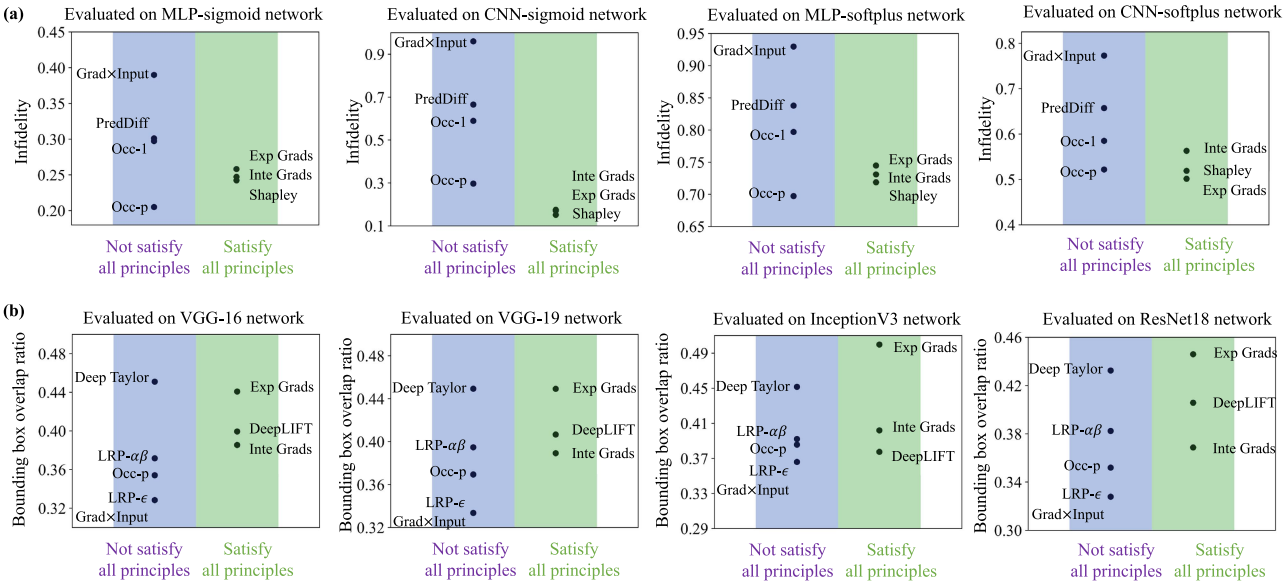


Fig. 5. (a) In general, attribution methods satisfying all principles usually show a lower infidelity. (b) On average, attribution methods satisfying all principles perform better on the bounding box overlap ratio. .

the y-axis denotes the corresponding infidelity metric of the attribution method. Fig. 5(a) indicates that, in general, attribution methods that satisfy all the proposed principles usually show lower infidelity.

The results show a relative consistency between the proposed principles and the infidelity. However, they evaluate attribution methods from different perspectives. Attribution methods with low infidelity do not always satisfy the proposed principles, *i.e.*, allocate interaction effects in a faithful manner.

D. Alignment With Human Intuition

In this subsection, we examine whether an attribution method deemed faithful by our proposed principles also aligns with human intuition. Considering the lack of an established formulation for human intuition, we have followed [30] to adopt the bounding box human annotation as a proxy to roughly represent human-intuitive attribution explanations for classification, and use the bounding box overlap ratio metric to measure the alignment between the attribution method and human intuition.

Specifically, the bounding box overlap ratio metric is defined as follows. Given a specific attribution method and an input image, the locations of target objects in the image have been annotated by human users using a bounding box set $B \subseteq N$. Let m denote the number of pixels in the bounding box, *i.e.*, $m = |B|$. On the other hand, we can also select a subset M of m pixels with the largest absolute attribution scores from all pixels in the image, *i.e.*, $M = \{i|rank(|a_i|) \leq m\}$. The overlap ratio metric measures the ratio of pixels with the largest absolute attribution scores (the most salient pixels) located within the bounding box.

$$\text{Overlap}(B, M) = |B \cap M|/|B| \tag{51}$$

A high bounding box overlap ratio indicates that the attribution method can well localize the target object like humans, *i.e.*, the attribution score aligns well with human intuition. For evaluation, we only test images on the ImageNet dataset [7] whose bounding box covers less than 33% pixels of the whole image, *i.e.*, $m < 33\%n$. We evaluate the bounding box accuracies of each attribution method on four different network designs, including VGG16 [34], VGG19 [34], InceptionV3 [38], and ResNet18 [20] networks.

Fig. 5(b) shows the relationship between the proposed principles and the bounding box overlap ratio. The x-axis denotes whether a specific attribution method satisfies all proposed principles, and the y-axis denotes the corresponding bounding box overlap ratio of the attribution method. Fig. 5(b) indicates that, on average, attribution methods, which satisfy the proposed three principles, also align well with human intuition, because they have relatively high bounding box overlap ratios. We have also observed that several attribution methods, which well fit human intuition, do not always satisfy all proposed principles. This is not surprising, because no evidence suggests that a DNN encodes the inference logic in the same way as human intuition.

V. Conclusion and Discussion

In this study, we propose the Taylor interaction effect as a unified perspective to explain the common mechanisms of fourteen attribution methods. Specifically, we prove that the attribution score estimated by each method can all be reformulated as a specific re-allocation of the Taylor independent effects and the Taylor interaction effects. Furthermore, from the unified perspective, we propose three principles for faithful attributions and then use them to evaluate the fourteen attribution methods.

Although the proposed Taylor interaction system is general for unifying various post-hoc attribution methods, there

are several attribution methods that cannot be unified into the system.

(i) The post-hoc attribution must have an explicit analytic formulation. Otherwise, the post-hoc method is difficult to be unified into the proposed system. For example, the guided feature inversion method [11], where attribution results are generated by optimizing a non-convex objective function and thus lack a unique and explicit analytic formulation, cannot be unified into the proposed system. The difficulty arises because our core theoretical proofs aim to prove the equivalence of the analytic forms between the original attribution proposed in previous papers and the attribution reformulated in our paper. If the original attribution lacks an explicit analytic formulation, proving such an equivalence becomes infeasible.

(ii) The post-hoc attribution should be presented as an n-dimensional numerical vector, where each dimension reflects the scalar importance score of a specific input variable. Otherwise, the post-hoc method is difficult to be unified into the proposed system. For example, the meaningful perturbation method [15], whose attribution result identifies the most influential subset of input variables, cannot be unified into the proposed system. The difficulty arises because the unified paradigm of the reformulated attribution in the proposed system is an n-dimensional numerical vector, where each dimension is defined as a weighted sum of scalar Taylor interaction effects. If the original attribution has a different form from the reformulated attribution, it is infeasible to prove the equivalence between them.

REFERENCES

[1] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9505–9515.

[2] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–16.

[3] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS One*, vol. 10, no. 7, 2015, Art. no. e0130140.

[4] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, 2010.

[5] H. Deng, N. Zou, W. Chen, G. Feng, M. Du, and X. Hu, "Mutual information preserving back-propagation: Learn to invert for faithful attribution," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 258–268.

[6] H. Deng, N. Zou, M. Du, W. Chen, G. Feng, and X. Hu, "A unified taylor framework for revisiting attribution methods," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11462–11469.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[8] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[9] M. Dikmen and C. M. Burns, "Autonomous driving in the real world: Experiences with tesla autopilot and summon," in *Proc. 8th Int. Conf. Automot. User Interfaces Interactive Veh. Appl.*, 2016, pp. 225–228.

[10] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, pp. 68–77, 2019.

[11] M. Du, N. Liu, Q. Song, and X. Hu, "Towards explanation of DNN-based prediction with guided feature inversion," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1358–1367.

[12] G. Erion, J. D. Janizek, P. Sturmfels, S. Lundberg, and S.-I. Lee, "Learning explainable models using attribution priors". 2019, *arXiv:1906.10670*.

[13] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S.-I. Lee, "Improving performance of deep learning models with axiomatic attribution priors and expected gradients," *Nature Mach. Intell.*, vol. 3, no. 7, pp. 620–631, 2021.

[14] R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 2950–2958.

[15] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 3429–3437.

[16] W. Fu, M. Wang, M. Du, N. Liu, S. Hao, and X. Hu, "Differentiated explanation of deep neural networks with skewed distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 2909–2922, Jun. 2022.

[17] M. Grabisch and M. Roubens, "An axiomatic approach to the concept of interaction among players in cooperative games," *Int. J. Game Theory*, vol. 28, no. 4, pp. 547–565, 1999.

[18] J. C. Harsanyi, "A simplified bargaining model for the n-person cooperative game," *Int. Econ. Rev.*, vol. 4, no. 2, pp. 194–220, 1963.

[19] S. Hart, "Shapley value," in *Game Theory*, Berlin, Germany: Springer, 1989, pp. 210–216.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[21] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9737–9748.

[22] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.

[23] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," *Pattern Recognit.*, vol. 65, pp. 211–222, 2017.

[24] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, 2018.

[25] W. Nie, Y. Zhang, and A. Patel, "A theoretical explanation for perplexing behaviors of backpropagation-based visualizations," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 3809–3818.

[26] H. Pei, B. Yang, J. Liu, and K. Chang, "Active surveillance via group sparse Bayesian learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1133–1148, Mar. 2022.

[27] J. Ren, M. Li, Q. Ren, H. Deng, and Q. Zhang, "Towards axiomatic, hierarchical, and symbolic explanation for deep models," 2021, *arXiv:2111.06206*.

[28] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should i trust you?" explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.

[29] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Toward interpretable machine learning: Transparent deep neural networks and beyond," 2020, *arXiv: 2003.07631*.

[30] K. Schulz, L. Sixt, F. Tombari, and T. Landgraf, "Restricting the flow: Information bottlenecks for attribution," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–18.

[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

[32] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 3145–3153.

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

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[35] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014, *arXiv:1412.6806*.

[36] M. Sundararajan, K. Dhamdhere, and A. Agarwal, "The shapley Taylor interaction index," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 9259–9268.

[37] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[39] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar, "On the (in) fidelity and sensitivity of explanations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10967–10978.
[40] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 818–833.
[41] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing DNN decisions: Prediction difference analysis," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–12.

Huiqi Deng received the PhD degree in applied mathematics from Sun Yat-sen University, China, in 2021. She is currently a postdoctoral researcher with Shanghai Jiao Tong University, China. She has previously visited HongKong Baptist University and Texas A&M University. Her research interests cover a wide range of explainable machine learning and adversarial robustness.

Na Zou is an assistant professor with University of Houston. Her research covers fair and interpretable machine learning, transfer learning, network modeling and inference. She has published papers with prestigious journals, such as *Technometrics*, *IISE Transactions* and *ACM Transactions*, including one Best Paper Finalist and one Best Student Paper Finalist at INFORMS QSR section and two featured articles at ISE Magazine. She was the recipient of IEEE Irv Kaufman Award and Texas A&M Institute of Data Science Career Initiation fellow.

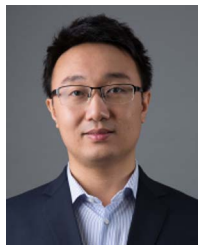
Mengnan Du received the PhD degree in computer science from Texas A&M University. He is an Assistant Professor with the New Jersey Institute of Technology (NJIT). He has previously worked/interned with Microsoft Research, Adobe Research, Intel, Baidu Research, Baidu Search Science and JD Explore Academy. His research covers a wide range of trustworthy machine learning, such as model explainability, fairness, and robustness. He has published more than 40 papers in prestigious venues such as NeurIPS, AAAI, KDD, WWW, ICLR, and ICML. He received more than 2,300 citations with an H-index of 16.

Guocan Feng received the PhD degree in computer science from Hong Kong Baptist University. He is a professor with Sun Yat-sen University, China. He was a research fellow with Digital Media Lab in University of Glamorgan and Univ. of Bradford in the U.K. His research interests include digital image processing, pattern recognition, computer vision, image retrieval and indexing, and manifold learning. He has published more than 80 refereed papers at conferences and journals.

Weifu Chen received the PhD degree in computing mathematics from Sun Yat-sen University China, in 2012. He is an associate professor with the Department of Computer Science, Guangzhou Maritime University. He was a senior research associate with the City University of Hong Kong from 2012 to 2016. From 2016 to 2022, he was an associate research fellow with Sun Yat-sen University. His research interests include pattern recognition and medical image processing.

Zheyang Li received the MSc degree from Shanghai JiaoTong University, Shanghai, China, in 2015. He is an algorithm researcher with Hikvision Research Institute. His research interests include perception algorithm, neural network acceleration, explainable AI.

Ziwei Yang received the MSc degree from Tianjin University, China, in 2018. He is an algorithm researcher with Hikvision Research Institute. His research interests mainly include neural architecture search, transfer learning and explainable machine learning.



Quanshi Zhang (Member, IEEE) received the PhD degree from the University of Tokyo, in 2014. He is an associate professor with Shanghai Jiao Tong University, China. From 2014 to 2018, he was a post-doctoral researcher with the University of California, Los Angeles. His research interests are mainly machine learning and computer vision. In particular, he has made influential research in explainable AI (XAI). He won the ACM China Rising Star Award at ACM TURC 2021. He is the speaker of the tutorials on XAI at IJCAI 2020 and IJCAI 2021. He was the co-chairs of the workshops towards XAI in ICML 2021, AAAI 2019, and CVPR 2019.