

# TOWARDS A GAME-THEORETIC VIEW OF BASELINE VALUES IN THE SHAPLEY VALUE

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

This paper aims to formulate the problem of estimating optimal baseline values, which are used to compute the Shapley value in game theory. In the computation of Shapley values, people usually set an input variable to its baseline value to represent the absence of this variable. However, there are no studies on how to ensure that baseline values represent the absence states of variables without bringing in additional information, which ensures the trustworthiness of the Shapley value. To this end, previous studies usually determine baseline values in an empirical manner, which are not reliable. Therefore, we revisit the feature representation of a deep model in game theory, and formulate the absence state of an input variable. From the perspective of game-theoretic interaction, we learn the optimal baseline value of each input variable. Experimental results have demonstrated the effectiveness of our method. *The code will be released when the paper is accepted.*

## 1 INTRODUCTION

Deep neural networks (DNNs) have exhibited significant success in various tasks, but the black-box nature of DNNs makes it difficult for people to understand the internal behavior of the DNN. Many methods have been proposed to explain the DNN, *e.g.* visualizing appearance patterns encoded by deep models (Simonyan et al., 2013; Yosinski et al., 2015; Mordvintsev et al., 2015), inverting features to the network input (Dosovitskiy & Brox, 2016), extracting receptive fields of neural activations (Zhou et al., 2015), and estimating the attribution/saliency/importance of input variables (*e.g.* pixels in an image, words in a sentence) *w.r.t.* the output of the model/network (Zhou et al., 2016; Selvaraju et al., 2017; Lundberg & Lee, 2017).

When we input a sample to a deep model, we focus on studies of estimating the attribution/saliency/importance of input variables to the model output. To this end, the Shapley value is widely used and considered as an unbiased measure of an input variable’s attribution (Shapley, 1953; Grabisch & Roubens, 1999; Lundberg & Lee, 2017; Sundararajan & Najmi, 2020). As an attribution metric, the Shapley value satisfies the *linearity*, *nullity*, *symmetry*, and *efficiency* axioms, which ensure the trustworthiness of the attribution.

However, the determination of baseline values is a typical problem with the theoretical foundation of the Shapley value, which hurts the trustworthiness of the explanation. As Figure 1 (left) shows, the Shapley value of an input variable is computed as the marginal difference of the model output between the case of maintaining this variable and the case of removing this variable. The removal of an input variable is usually implemented by setting this variable to a certain baseline value (or called reference value), to represent its absence state. In previous studies, people usually simply set the removed input variables to zero or the mean value over different inputs (Ancona et al., 2019; Dabkowski & Gal, 2017), namely the *zero baseline* and the *mean baseline*. Fong & Vedaldi (2017) blurred the input image and used the smoothed pixel values as baseline values, namely the *blurring baseline*. Besides, instead of setting specific baseline values, Covert et al. (2020b); Frye et al. (2021) determined baseline values conditionally depending on the neighboring contexts, namely the *conditional baseline*.

**Problem with empirical settings of baseline values.** Frye et al. (2021) has pointed out that the incorrect setting of baseline values may lead to dramatically incorrect attributions. We also find that incorrect baseline values will mistakenly explain a single complex concept as the mixture of massive simple concepts (see Appendix E.1). In fact, the essence of the problem with baseline values can be summarized as that baseline values do not satisfy the following two requirements. First, baseline

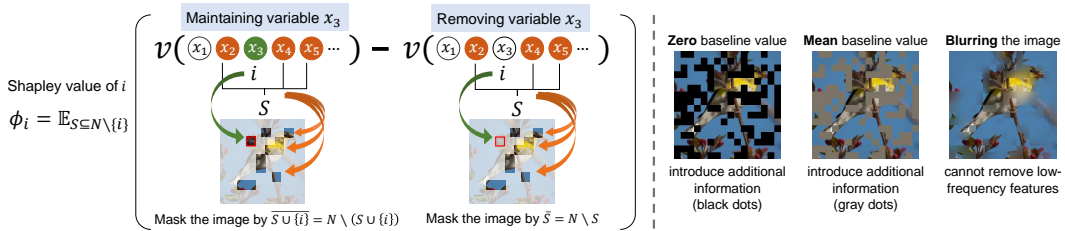


Figure 1: (Left) The computation of Shapley values. (Right) Previous settings of baseline values, including the *zero baseline*, *mean baseline*, and *blurring baseline*.

values should remove all information represented by original variable values. Second, baseline values should not bring in new/abnormal information. Otherwise, the computed Shapley value may not faithfully reflect the attribution of input variables.

Unfortunately, all the aforementioned baseline values have the above problem. As Figure 1 (right) shows, the *zero baseline* and *mean baseline* would introduce new/abnormal information. The *blurring baseline* cannot remove all signals. Besides, the *conditional baseline* cannot be considered to faithfully satisfy the linearity and nullity axioms from some aspects, and they are not suitable for continuous variables due to the computational cost. Detailed analysis is introduced in Section 3.1.

**Solutions.** In order to solve the above issue in both theory and practice, we define the absence state of input variables in game theory, *i.e.* the same theoretic system of defining the Shapley value. Given a trained model and input samples, we aim to learn baseline values for different input variables, which satisfy the following two requirements. (1) Our method is supposed to retain all the four axioms of Shapley values, which ensure the solidness of the Shapley value theory. (2) The baseline value removes the information of the original input variable without bringing in new information.

In order to define the absence state of an input variable, let us first revisit the feature representation of a model in terms of game-theoretic interactions. Let an input sample of a deep model has  $n$  variables  $N = \{1, 2, \dots, n\}$ . In the deep model, input variables do not contribute to the model output individually. Instead, different input variables cooperate with each other to form some interaction patterns for inference. Thus, each subset of input variables  $S \subseteq N$  can be considered as a potential interaction pattern. To this end, we discover that we can use the Harsanyi dividend (Harsanyi, 1963)  $I(S)$  to measure the benefit of the interaction between variables in the subset/interaction pattern  $S \subseteq N$ . In this way, the output of the model can be decomposed as the sum of  $2^n - 1$  interaction patterns, *i.e.*  $model\ output = \sum_{S \subseteq N, S \neq \emptyset} I(S) + constant$ , as shown in Figure 6. Let us consider a toy example where a model can be explained to contain two interaction patterns  $f(x) = I(\{x_1, x_2\}) + I(\{x_2, x_3, x_4\}) = w_{12}\delta_1\delta_2 + w_{234}\delta_2\delta_3\delta_4$ .  $\delta_i \in \{0, 1\}$  represents the presence/absence of the variable  $x_i$ . Each interaction pattern represents an AND relationship between multiple input variables, *e.g.*  $I\{x_1, x_2\} = w_{12}\delta_1\delta_2 \neq 0$  if and only if  $(\delta_1 = 1) \& (\delta_2 = 1)$ .

The interaction pattern provides us a game-theoretic way to model the absence state of an input variable. Specifically, if an interaction pattern  $S$  has a large absolute value  $|I(S)|$ , we consider the interaction between variables in  $S$  has a significant influence on inference. Such interaction patterns are called *salient patterns*. Thus, we can count the number of salient patterns  $S$  associated with the variable  $i \in S$  as the importance of  $i$ . In this way, if an input variable is not involved in any salient patterns, then this variable is negligible for inference, thus can be considered absent. However, it is difficult to make an input variable not be involved in any salient patterns. Therefore, the absence state of an input variable is defined as the baseline value that makes the variable  $i$  be involved in the least salient patterns. In addition, the computational cost of learning such optimal baseline values is exponential. Fortunately, we discover an approximate yet efficient solution to it. Therefore, by deactivating most salient interaction patterns, the learned baseline value removes most information from the input variable without bringing in new information.

**Contributions** of this paper can be summarized as follows. (1) We formulate two requirements for baseline values, and define the optimal baseline value in game theory. (2) We develop a method to estimate optimal baseline values, which ensures the reliability and trustworthiness of the Shapley value. (3) We measure the multi-variate interaction between input variables based on the Harsanyi dividend, and prove its theoretical connections to other attributions and interactions, which demonstrate the rigorousness of this metric.

## 2 RELATED WORKS

**Shapley values.** The Shapley value (Shapley, 1953) was first proposed in game theory, which was considered as an unbiased distribution of the overall reward in a game to each player. Some previous studies used the Shapley value to explain different models (Grömping, 2007; Štrumbelj et al., 2009; Lundberg & Lee, 2017). Sundararajan et al. (2017) proposed Integrated Gradients based on the AumannShaple (Aumann & Shapley, 2015) cost-sharing technique. Besides the above local explanations, Covert et al. (2020b) focused on the global interpretability. However, the computational cost of Shapley value is large. Therefore, Lundberg & Lee (2017); Lundberg et al. (2018); Aas et al. (2019); Ancona et al. (2019) explored different approximate yet efficient estimations of Shapley values to speed up the computation.

Some previous studies also discussed problems with baseline values in the computation of Shapley values. Most studies (Covert et al., 2020a; Merrick & Taly, 2020; Sundararajan & Najmi, 2020; Kumar et al., 2020) compared influences of baseline values on explanations, without providing any principle rules for setting baseline values. Besides, Agarwal & Nguyen (2019) and Frye et al. (2021) used generative models to alleviate the out-of-distribution problem caused by baseline values.

Unlike previous studies, we rethink and formulate baseline values from the perspective of game theory. We define the absence state of input variables based on the multi-variate interaction, and further propose a method to learn optimal baseline values.

**Interactions.** Interactions between input variables of deep models have been widely investigated in recent years. Many people defined interactions between input variables or weights to explain different models from different perspectives (Sorokina et al., 2008; Tsang et al., 2018; Murdoch et al., 2018; Singh et al., 2018; Jin et al., 2019; Cui et al., 2019). In game theory, Grabisch & Roubens (1999) proposed the Shapley interaction index based on Shapley values. Janizek et al. (2020) extended the Integrated Gradients method (Sundararajan et al., 2017) to explain pairwise feature interactions in DNNs. Sundararajan et al. (2020) defined the Shapley-Taylor index to measure interactions over binary features. In this paper, we use the multi-variate interaction of input variables based on the Harsanyi dividend (Harsanyi, 1963). Our interaction metric has strong connections to (Grabisch & Roubens, 1999), but represents elementary interaction patterns in a more detailed manner, and satisfies the efficiency axiom.

## 3 LEARNING BASELINE VALUES FOR SHAPLEY VALUES

**Preliminaries: Shapley values.** The Shapley value (Shapley, 1953) was first introduced in the game theory, which measures the importance/contribution/attribution of each player in a game. Let us consider a game with multiple players. Each player participates in the game and receives a reward individually. Some players may form a coalition and play together to pursue a higher reward. Different players in the game usually contribute differently to the game, and then the question is *how to fairly assign the total reward in the game to each player*. To this end, the Shapley value is considered as a unique unbiased approach that fairly allocates the reward to each player (Weber, 1988; Lundberg & Lee, 2017; Sundararajan & Najmi, 2020).

Given a game  $v$  with  $n$  players, let  $N = \{1, 2, \dots, n\}$  denote the set of all players and  $2^N = \{S | S \subseteq N\}$  denote all subsets of players. The game  $v : 2^N \mapsto \mathbb{R}$  is represented as a function that maps a subset of players  $S \subseteq N$  to a scalar reward  $v(S) \in \mathbb{R}$ , *i.e.* the reward gained by all players in  $S$ . Specifically,  $v(\emptyset)$  represents the baseline reward without any players. Considering the player  $i \notin S$ , if the player  $i$  joins in  $S$ ,  $v(S \cup \{i\}) - v(S)$  is considered as the marginal contribution of  $i$ . Thus, the Shapley value of the player  $i$  is computed as the weighted marginal contribution of  $i$  *w.r.t.* all subsets of players  $S \subseteq N \setminus \{i\}$ , as follows.

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} p(S) [v(S \cup \{i\}) - v(S)], \quad p(S) = \frac{|S|!(n - |S| - 1)!}{n!} \quad (1)$$

The fairness of the Shapley value is ensured by the following four axioms (Weber, 1988).

(a) *Linearity axiom:* If two games can be merged into a new game  $u(S) = v(S) + w(S)$ , then the Shapley values of the two old games also can be merged, *i.e.*  $\forall i \in N, \phi_{i,u} = \phi_{i,v} + \phi_{i,w}; \forall c \in \mathbb{R}, \phi_{i,c \cdot u} = c \cdot \phi_{i,u}$ .

Table 1: Analysis about previous choices of baseline values.

Setting of baseline values	Baseline values are constant or not	Different samples share the same baseline values or not	Shortcomings
Zero (Ancona et al., 2019) (Sundararajan et al., 2017)	✓	✓	introduce additional information
Mean values (Dabkowski & Gal, 2017)	✓	✓	introduce additional information
Blurring (Fong & Vedaldi, 2017) (Fong et al., 2019)	✓	✗	cannot remove low-frequency components
Marginalize (marginal distribution) (Lundberg & Lee, 2017)	✗	✓	assume feature independence
Marginalize (conditional distribution) (Covert et al., 2020b; Frye et al., 2021)	✗	✗	destroy linearity and nullity axioms (Sundararajan & Najmi, 2020)

(b) *Dummy axiom and nullity axiom*: The dummy player  $i$  is defined as a player without any interactions with other players, *i.e.* satisfying  $\forall S \subseteq N \setminus \{i\}, v(S \cup \{i\}) = v(S) + v(\{i\})$ . Then, the dummy player’s Shapley value is computed as  $\phi_i = v(\{i\})$ . The null player  $i$  is defined as a player that satisfies  $\forall S \subseteq N \setminus \{i\}, v(S \cup \{i\}) = v(S)$ . Then, the null player’s Shapley value is  $\phi_i = 0$ .

(c) *Symmetry axiom*: If  $\forall S \subseteq N \setminus \{i, j\}, v(S \cup \{i\}) = v(S \cup \{j\})$ , then  $\phi_i = \phi_j$ .

(d) *Efficiency axiom*: The overall reward of the game is equal to the sum of Shapley values of all players, *i.e.*  $v(N) - v(\emptyset) = \sum_{i \in N} \phi_i$ .

**Using Shapley values to explain deep models.** Given a trained model  $f : \mathbb{R}^n \mapsto \mathbb{R}$  and an input sample  $x \in \mathbb{R}^n$ , we can consider each input variable (*e.g.* a dimension, a pixel, or a word)  $x_i$  as a player ( $i \in N$ ), and consider the deep model as a game. The model output  $f(x)$  is regarded as the reward  $v(N)$ . Thus, the Shapley value  $\phi_i$  measures the attribution of the  $i$ -th variable  $x_i$  *w.r.t.* the model output. In this case,  $v(S)$  represents the model output, when variables in  $S$  are present and variables in  $\bar{S} = N \setminus S$  are absent. People usually set the variables in  $\bar{S} = N \setminus S$  to their baseline values, in order to represent their absence states. In this way,  $v(S)$  can be represented as follows.

$$v(S) = f(\text{mask}(x, S)), \quad \text{mask}(x, S) = x_S \sqcup b_{\bar{S}}, \quad (x_S \sqcup b_{\bar{S}})_i = \begin{cases} x_i, & i \in S \\ b_i, & i \in \bar{S} = N \setminus S \end{cases} \quad (2)$$

where  $\text{mask}(x, S)$  denotes the masked sample, and  $b_i$  denotes the baseline value of the  $i$ -th input variable.  $\sqcup$  indicates the concatenation of  $x$ ’s dimensions in  $S$  and  $b$ ’s dimensions in  $\bar{S} = N \setminus S$ .

### 3.1 PROBLEMS WITH SHAPLEY VALUES

According to Equations (1) and (2), the Shapley value computes the marginal contribution of the variable  $i$ ,  $v(S \cup \{i\}) - v(S)$ , under different masked contexts  $S \subseteq N$ . All input variables not in  $S$  are set to their baseline values to represent their absence states. Therefore, a good baseline value should satisfy the following two requirements. (1) First, it removes the information represented by the original value  $x_i$ . (2) Second, it does not introduce additional information to the input.

Otherwise, incorrect baseline values may lead to dramatically incorrect attributions, and mistakenly explain a single complex concept as the mixture of massive simple concepts, as discussed in Section 3.3. To this end, let us discuss existing empirical settings for baseline values (see Table 1).

- *Mean baseline values.* The baseline value of each input variable is set to the mean value of this variable over all samples (Dabkowski & Gal, 2017), *i.e.*  $b_i = \mathbb{E}_x[x_i]$ . This method actually introduces additional information to the input. As Figure 1 (right) shows, setting pixels to mean baseline values brings in massive additional gray dots to the image, rather than represent absence states of variables.

- *Zero baseline values.* Baseline values of all input variables are set to zero (Ancona et al., 2019; Sundararajan et al., 2017), *i.e.*  $\forall i \in N, b_i = 0$ . As Figure 1 (right) shows, just like mean baseline values, zero baseline values also introduce additional information (black dots) to the input. Note that because pixels in the input image are usually normalized to zero mean and a unit variance, the setting of zero baseline values is equivalent to the setting of mean baseline values in this case.

- *Blurring input samples.* Fong & Vedaldi (2017) and Fong et al. (2019) remove variables in the input image by blurring each input variable  $x_i (i \in \bar{S} = N \setminus S)$  using a Gaussian kernel. In this case, different input samples may have different baseline values. This approach can only remove high-frequency signals, but fails to remove low-frequency signals (Covert et al., 2020a; Sturmfels et al., 2020).

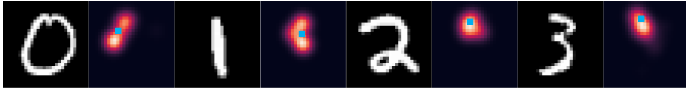


Figure 2: Visualization of contextual pixels  $j$  that collaborates with a certain pixel  $i$  (the blue dot).

- For each input variable, determining a different baseline value for this variable given each specific context  $S$  (neighboring variables). Instead of fixing baseline values as constants, some studies use varying baseline values, which are determined temporarily by the context  $S$  in a specific sample  $x$ , to compute  $v(S|x)$  given  $x$ . Some methods (Frye et al., 2021; Covert et al., 2020b) define  $v(S|x)$  by modeling the conditional distribution of variable values in  $\bar{S} = N \setminus S$  given the context  $S$ , i.e.  $v(S|x) = \mathbb{E}_{p(x'|\bar{x}_S)}[f(x_S \sqcup x'_S)]$ . However, these methods apply varying baseline values based on the dependency between input variables, which do not faithfully satisfy the linearity axiom and the nullity axiom of the Shapley value from some aspects (please see (Sundararajan & Najmi, 2020) or Appendix C for proof). Moreover, these methods compute a specific conditional distribution  $p(x'|\bar{x}_S)$  for each of  $2^n$  contexts  $S$  with a very high computational cost. By assuming that input variables are independent with each other, Lundberg & Lee (2017) simplify the above conditional baseline values to the marginal baseline values, i.e.  $v(S|x) = \mathbb{E}_{p(x')}[f(x_S \sqcup x'_S)]$ . However, the computational cost of marginal baseline values is still very large, thus being not suitable for continuous variables.

### 3.2 MULTI-VARIATE INTERACTIONS AND ABSENCE STATES OF INPUT VARIABLES

**Multi-variate interactions.** In this study, we use the Harsanyi dividend (Harsanyi, 1963) to measure the interaction benefit, which is the foundation of representing the presence of an input variable. Given a trained model and the input sample  $x$  with  $n$  variables  $N = \{1, 2, \dots, n\}$ , let  $v(S)$  denote the model output when only variables in  $S \subseteq N$  are input into the model, according to Equation (2). Then,  $v(N) - v(\emptyset)$  represents the overall inference benefit of the model output owing to all input variables in  $x$ , w.r.t. the model output without given any variables. In a deep model, different input variables do not contribute to the model output individually. Instead, they interact with each other to form interaction patterns for inference. We discover that when we use the Harsanyi dividend (Harsanyi, 1963) to quantify the benefit  $I(S)$  from the interaction between variables in an interaction pattern  $S$ , the overall benefit  $v(N) - v(\emptyset)$  can be decomposed into the sum of benefits  $I(S)$  of different interaction patterns  $S \subseteq N$ .

$$v(N) - v(\emptyset) = \sum_{S \subseteq N, S \neq \emptyset} I(S) \quad (3)$$

For an interaction pattern  $S \subseteq N$ , if  $I(S) > 0$ , the collaboration between variables in  $S$  has positive effects on model output. If  $I(S) < 0$ , the collaboration has negative effects. If  $I(S) \approx 0$ , variables in  $S$  do not have collaborations. The Harsanyi dividend is defined to measure the additional benefit from the collaboration of input variables in  $S$ , in comparison with the benefit when they work individually and when they form smaller patterns. Specifically,  $v(S) - v(\emptyset)$  denotes the overall benefit from all variables in  $S$ , and then we remove the marginal benefits owing to collaborations of all smaller subsets  $L$  of variables in  $S$ , i.e.  $\{I(L)|L \subsetneq S, L \neq \emptyset\}$ , as follows.

$$I(S) = \underbrace{v(S) - v(\emptyset)}_{\text{the benefit from all variables in } S} - \sum_{L \subsetneq S, L \neq \emptyset} I(L) \Rightarrow I(S) = \sum_{L \subseteq S} (-1)^{|S|-|L|} v(L) \quad (4)$$

Let us consider the example when a model uses  $y_{\text{head}} = \text{sigmoid}(x_{\text{eyes}} + x_{\text{nose}} + x_{\text{mouth}} + x_{\text{ears}} - \text{constant})$  to represent the AND relationship  $y_{\text{head}} \leftarrow (x_{\text{eyes}}) \& (x_{\text{nose}}) \& (x_{\text{mouth}}) \& (x_{\text{ears}})$ . In this case, the interaction benefit  $I(\{x_{\text{eyes}}, x_{\text{nose}}, x_{\text{mouth}}\})$  measures the marginal benefit when *eyes*, *nose*, and *mouth* collaborate with each other, where the benefits from smaller subsets (e.g. the interaction between *eyes* and *nose*) and the individual benefits (e.g. the individual benefit from *nose*) are removed. In other words,  $I(\{x_{\text{eyes}}, x_{\text{nose}}, x_{\text{mouth}}\}) = v(\{x_{\text{eyes}}, x_{\text{nose}}, x_{\text{mouth}}\}) - v(\emptyset) - I(\{x_{\text{eyes}}\}) - I(\{x_{\text{nose}}\}) - I(\{x_{\text{mouth}}\}) - I(\{x_{\text{eyes}}, x_{\text{nose}}\}) - I(\{x_{\text{eyes}}, x_{\text{mouth}}\}) - I(\{x_{\text{nose}}, x_{\text{mouth}}\})$ . Furthermore, Figure 2 visualizes the distribution of contextual pixels  $j$  that collaborate with a certain pixel  $i$  (the blue dot). Please see Appendix E.2 for details of the visualization method.

**Properties of multi-variate interactions.** We extend the *linearity*, *dummy*, *symmetry* axioms of Shapley values to the above definition of  $I(S)$  (please see the Appendix B.1 for details). Besides, the above metric has also been proven to have a close relationship to the Shapley value. We further prove its relationship to other game-theoretic interaction metrics in Appendix B.3, including the Shapley

interaction index (Grabisch & Roubens, 1999) and the Shapley Taylor interaction index (Sundararajan et al., 2020).

**Counting salient patterns associated with each input variable.** According to Equation (3), some interaction patterns have large absolute values  $|I(S)|$ , which have significant influences on the model output, namely *salient patterns*. In comparison, other patterns have small absolute values  $|I(S)|$ , having little effects on the model output, namely *noisy patterns*. According to (Harsanyi, 1963), the benefit of an interaction pattern consisting of  $m$  variables can be fairly assigned to the  $m$  variables. In this way, for each input variable  $i \in N$ , we can consider the number of salient interaction patterns associated with this variable as the numerical importance of this variable to the model output.

**Definition of absence state and optimal baseline values.** The computation of the Shapley value is conducted based on the assumption that baseline values represent the absence of input variables. The number of salient patterns associated with an input variable  $i$  indicates the importance (presence) of  $i$ , but it is difficult to find a baseline value that removes *all* salient patterns associated with  $i$ . Thus, the absence state of the variable  $i$  can be achieved, when the baseline value of  $i$  removes *most* existing salient patterns associated with  $i$  without triggering new salient patterns. In other words, baseline values are learned to remove the original information from input variables and to avoid bringing in new information. Therefore, the learning of the baseline value  $\hat{b}_i$  of the input variable  $i$  is formulated to sparsify the salient patterns associated with  $i$ .

$$\hat{b}_i = \arg \min_{b_i} \sum_{S \subseteq N \setminus \{i\}} \mathbb{1}_{|I(S \cup \{i\})| \geq \tau} \quad (5)$$

where  $\tau$  denotes the threshold to determine salient patterns.

### 3.3 ESTIMATING BASELINE VALUES TO MINIMIZE THE NUMBER OF SALIENT PATTERNS

Equation (5) guides the learning of optimal baseline values, but the computational cost of enumerating/counting all salient patterns is exponential. Thus, we need to find an approximate solution to learning baseline values. Fortunately, the *order* of interaction patterns provides us with a new perspective to solve this problem. The *order* of the interaction benefits  $I(S)$  is defined as the cardinality of  $S$ , *i.e.* the order  $m = |S|$ . To this end, we obtain the following two propositions.

(1) *Correct baseline values ensure sparse high-order interaction patterns.* Theoretical analysis and preliminary experiments (see Table 6 in Appendix E.1) have shown that incorrect baseline values usually explain a high-order interaction utility (the collaboration between massive variables) as the sum of massive low-order interaction utilities, which are actually unnecessary. For example, given a certain set of baseline values, the model may be explained to contain a single high-order interaction pattern between massive input variables. However, the same model may be explained to encode massive low-order interactions between a few baseline values, given another set of baseline values. Thus, we aim to learn baseline values that correctly reflect the logic of the model by using sparse, salient, and high-order interaction patterns.

(2) *High-order interaction patterns are more likely to be deactivated.* From another perspective, the benefit of each interaction pattern  $I(S)$  represents an AND relationship between all variables in  $S$ . The absence of any variable (*i.e.* setting this variable to its baseline value) deactivates this pattern. Obviously, high-order interaction patterns depend on massive variables. Thus, high-order interaction patterns are more likely to be deactivated when some variables are masked during the computation of Shapley values. If the model is mainly represented by high-order interaction patterns, it is more likely to deactivate patterns and achieve the minimum pattern number.

Based on the above propositions, we aim to learn baseline values that represent the model using sparse and salient high-order interaction patterns. According to Equation (3), the weighted sum of high-order interactions and low-order interactions is relatively stable as  $v(N) - v(\emptyset)$ . It means that penalizing the influence of low-order interaction patterns will increase the influence of high-order interaction patterns, thereby pushing the model to mainly use high-order interaction patterns for inference. Therefore, *the objective of learning baseline values can be transformed to a loss function that penalizes the influence of low-order interactions*. In this way, baseline values are learned to strengthen high-order interaction patterns towards salient patterns by penalizing low-order interaction patterns towards noisy patterns.

**An approximate yet efficient solution.** Directly computing  $I(S)$  is NP-hard. Therefore, we design loss functions based on the following multi-order Shapley values and the multi-order marginal benefits

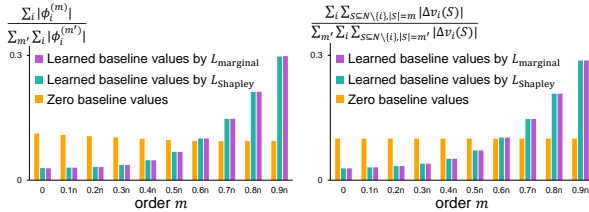


Figure 3: Our methods successfully boost the influence of high-order Shapley values and high-order marginal benefits, and reduce the influence of low-order Shapley values and low-order marginal benefits, computed on a function in (Tsang et al., 2018).

to penalize low-order interaction patterns, which boost the computational efficiency. We prove that the Shapley value  $\phi_i$  can be decomposed into Shapley values of different orders  $\phi_i^{(m)}$  ( $\phi_i = \frac{1}{n} \sum_{m=0}^{n-1} \phi_i^{(m)}$ ), as well as marginal benefits of different orders  $\Delta v_i(S)$  ( $\phi_i = \frac{1}{n} \sum_{m=0}^{n-1} \mathbb{E}_{S \subseteq N \setminus \{i\}, |S|=m} \Delta v_i(S)$ ), which are proved in Appendix D.  $\phi_i^{(m)}$  and  $\Delta v_i(S)$  are given as follows.

$$\begin{aligned} \phi_i^{(m)} &\stackrel{\text{def}}{=} \mathbb{E}_{\substack{S \subseteq N \setminus \{i\} \\ |S|=m}} [v(S \cup \{i\}) - v(S)] \Rightarrow \phi_i^{(m)} = \mathbb{E}_{\substack{S \subseteq N \setminus \{i\} \\ |S|=m}} \left[ \sum_{L \subseteq S} I(L \cup \{i\}) \right] \\ \Delta v_i(S) &\stackrel{\text{def}}{=} v(S \cup \{i\}) - v(S) \Rightarrow \Delta v_i(S) = \sum_{L \subseteq S} I(L \cup \{i\}) \end{aligned} \quad (6)$$

The order of the marginal benefit  $\Delta v_i(S)$  is defined as  $m = |S|$ , the number of elements in  $S$ . For a low order  $m$ ,  $\phi_i^{(m)}$  denotes the attribution of the  $i$ -th input variable, when  $i$  cooperates with a few contextual pixels. For a high order  $m$ ,  $\phi_i^{(m)}$  corresponds to the impact of  $i$  when it collaborates with massive contextual variables. Similarly, the order of  $\Delta v_i(S)$  measures the marginal benefit of the  $i$ -th input variable to the model output with contexts composed of  $m = |S|$  variables.

Equation (6) has shown that high-order interaction patterns  $I(S)$  are only contained by high-order Shapley values  $\phi_i^{(m)}$  and high-order marginal benefits  $\Delta v_i(S)$ . In order to penalize the influence of low-order interaction patterns and boost high-order interaction patterns, we propose the following two loss functions to penalize the strength of low-order Shapley values,  $|\phi_i^{(m)}|$ , and to penalize the strength of low-order marginal benefits,  $|\Delta v_i(S)|$ , respectively.

$$L_{\text{Shapley}}(\mathbf{b}) = \sum_{m \sim \text{Unif}(0, \lambda)} \sum_{x \in X} \sum_{i \in N} |\phi_i^{(m)}|, \quad L_{\text{marginal}}(\mathbf{b}) = \sum_{m \sim \text{Unif}(0, \lambda)} \sum_{x \in X} \sum_{i \in N} \mathbb{E}_{S \subseteq N, |S|=m} |\Delta v_i(S)| \quad (7)$$

where  $\lambda > m$  denotes the maximum order to be penalized. Figure 3 shows the distribution of the ratio of  $\sum_i |\phi_i^{(m)}|$  and the ratio of  $\sum_i \sum_{S \subseteq N \setminus \{i\}, |S|=m} |\Delta v_i(S)|$  of different orders, which demonstrates that our methods effectively boost the influence of high-order interaction patterns. The loss function on marginal benefits is more fine-grained than the loss function on the multi-order Shapley value.

## 4 EXPERIMENTS

**Learning baseline values.** We used our method to learn baseline values for MLPs and LeNet (LeCun et al., 1998) trained on the UCI South German Credit dataset (Asuncion & Newman, 2007), the UCI Census Income dataset (Asuncion & Newman, 2007), and the MNIST dataset (LeCun et al., 1998), respectively. Based on the UCI datasets, we learned MLPs following settings in (Guidotti et al., 2018). We learned baseline values using either  $L_{\text{Shapley}}$  or  $L_{\text{marginal}}$  as the loss function. In the computation of  $L_{\text{Shapley}}$ , we set  $v(S) = \log \frac{p(y^{\text{truth}} | \text{mask}(x, S))}{1 - p(y^{\text{truth}} | \text{mask}(x, S))}$ . In the computation of  $L_{\text{marginal}}$ ,  $|\Delta v_i(S)|$  was set to  $|\Delta v_i(S)| = \|h(\text{mask}(x, S \cup \{i\})) - h(\text{mask}(x, S))\|_1$ , where  $h(\text{mask}(x, S))$  denoted the output feature of the second last layer given the masked input  $\text{mask}(x, S)$ , in order to boost the efficiency of learning. Please see Appendix E.3 for other potential settings of  $v(S)$ . We used two ways to initialize baseline values before the learning phase, *i.e.* setting to zero (Ancona et al., 2019; Sundararajan et al., 2017) or the mean values over different samples (Dabkowski & Gal, 2017), namely *zero-init* and *mean-init*, respectively. We set  $\lambda = 0.2n$  for the MNIST dataset, and set  $\lambda = 0.5n$  for the simpler data in two UCI datasets. We compared our methods with five previous choices of baseline values introduced in Section 3.1, *i.e.* zero baseline values (Ancona et al., 2019), mean baseline values (Dabkowski & Gal, 2017), blurring images (Fong & Vedaldi, 2017; Fong et al., 2019), and two implementations of varying baseline values in SHAP (Lundberg & Lee, 2017) and SAGE (Covert et al., 2020b). Zero baseline values, mean baseline values, blurring images, and our learned baseline values allowed us to compute the Shapley value using the sampling-based approximation (Castro et al., 2009). This

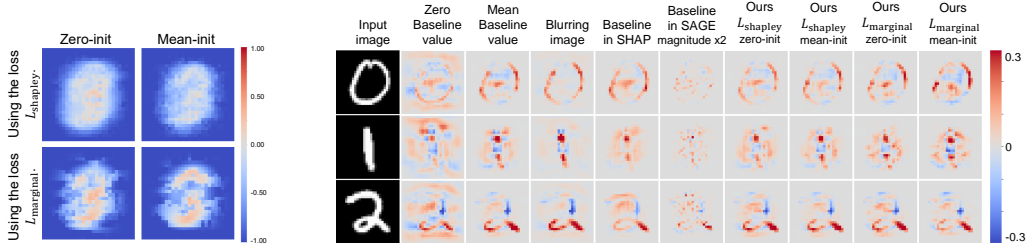


Figure 4: (Left) The learned baseline values on the MNIST dataset (better viewed in color). (Right) Shapley values produced with different baseline values on the MNIST dataset.

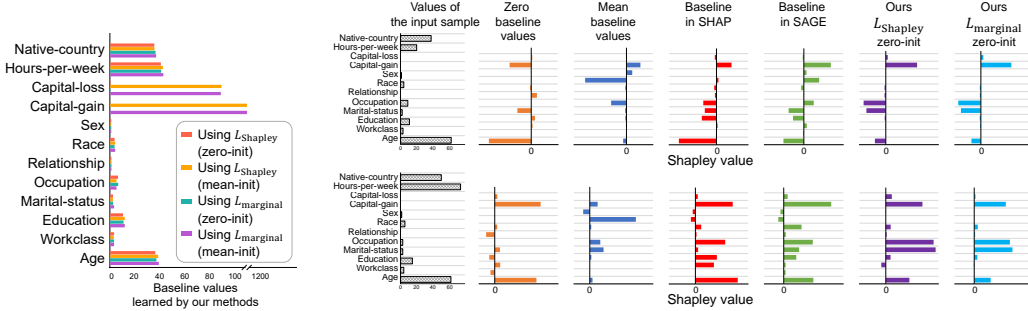


Figure 5: The learned baseline values (left) and Shapley values computed with different baseline values (right) on the UCI Census Income dataset. Results on the UCI South German Credit dataset are shown in Appendix E.4.

approximation ensured the unbiased estimation of Shapley values. The stability of the computed Shapley values with different sampling numbers was examined in Appendix E.5. For the varying baseline values, we used the code released by (Lundberg & Lee, 2017) and (Covert et al., 2020b).

Figure 4 (left) shows the learned baseline values on the MNIST dataset. Figure 4 (right) compares the computed Shapley values with different choices of baseline values. Compared to zero/mean/blurring baseline values, our learned baseline values removed noisy variables on the background, which were far from the digit in the image. Compared to the baseline values in SHAP, our method yielded more informative attributions. Shapley values computed using the baseline values in SAGE were dotted. In comparison, our method generated smoother attributions. Furthermore, our settings satisfied the linearity axiom and the nullity axiom. Figure 5 shows the learned baseline values and the computed Shapley values on the UCI Census Income dataset. Unlike our methods, attributions generated by the zero/mean baseline values conflicted with the results of all other methods.

**Verifying baseline values on synthetic functions.** People usually cannot determine the ground truth of baseline values for real images, such as the MNIST dataset. Therefore, we conducted experiments on synthetic functions with ground-truth baseline values, in order to verify the correctness of the learned baseline values. We randomly generated 100 functions whose interaction patterns and ground truth of baseline values could be easily determined. This dataset will be released after the paper acceptance. The generated functions were composed of addition, subtraction, multiplication, exponentiation, and the *sigmoid* operations (see Table 2). For example, for the function  $y = \text{sigmoid}(3x_1x_2 - 3x_3 - 1.5) - x_4x_5 + 0.25(x_6 + x_7)^2$ ,  $x_i \in \{0, 1\}$ , there were three salient interaction patterns (*i.e.*  $\{x_1, x_2, x_3\}$ ,  $\{x_4, x_5\}$ ,  $\{x_6, x_7\}$ ), which were activated only if  $x_i = 1$  for  $i \in \{1, 2, 4, 5, 6, 7\}$  and  $x_3 = 0$ . In this case, the ground truth of baseline values should be  $b_i^* = 0$  for  $i \in \{1, 2, 4, 5, 6, 7\}$  and  $b_3^* = 1$ . Please see Appendix E.6 for more discussions about the setting of ground-truth baseline values. We used our methods to learn baseline values on these functions and tested the accuracy. Note that  $|b_i - b_i^*| \in [0, 1]$ . If  $|b_i - b_i^*| < 0.5$ , we consider the learned baseline value correct; otherwise incorrect. We set  $\lambda = 0.5n$  in both  $L_{\text{Shapley}}$  and  $L_{\text{marginal}}$ . Experimental results are reported in Table 3 and are discussed later.

**Verifying baseline values on functions in (Tsang et al., 2018).** Besides, we also evaluated the correctness of the learned baseline values using functions proposed in (Tsang et al., 2018). Among all the 92 input variables in these functions, the ground truth of 61 variables could be determined (see Appendix E.6). Thus, we used these annotated baseline values to test the accuracy on these functions.



Table 2: Examples of generated functions and their ground-truth baseline values.

Functions ( $\forall i \in N, x_i \in \{0, 1\}$ )	The ground truth of baseline values
$-0.185x_1(x_2 + x_3)^{2.432} - x_4x_5x_6x_7$	$b_i^* = 0$ for $i \in \{1, 2, 3, 4, 5, 6, 7\}$
$-x_1x_2x_3 + \text{sigmoid}(-5x_4x_5x_6x_7 + 2.50) - x_8x_9$	$b_i^* = 1$ for $i \in \{4, 5, 6, 7\}$ , $b_i^* = 0$ for $i \in \{1, 2, 3, 8, 9\}$
$-\text{sigmoid}(+4x_1 - 4x_2 + 4x_3 - 6.00) - x_4x_5x_6x_7 - x_8x_9x_{10}$	$b_i^* = 1$ for $i = 2$ , $b_i^* = 0$ for $i \in \{1, 3, 4, 5, 6, 7, 8, 9, 10\}$

Table 3: Accuracy of the learned baseline values.

	$L_{\text{Shapley}}$			$L_{\text{marginal}}$		
	initialize with 0	initialize with 0.5	initialize with 1	initialize with 0	initialize with 0.5	initialize with 1
Synthetic functions	98.06%	98.70%	98.70%	98.06%	98.14%	98.14%
Functions in (Tsang et al., 2018)	88.52%	91.80%	90.16%	86.89%	91.80%	90.16%

Table 4: Accuracy of Shapley values on the extended Addition-Multiplication dataset when using different settings of baseline values.

	mean baseline		baseline values in SHAP	Ours
	baseline	baseline		
Accuracy	82.88%	72.63%	81.25%	100%

Table 5: The learned baseline values successfully recovered original samples from adversarial examples.

	$\ x^{\text{adv}} - x\ _2$	$\ \mathbf{b} - x\ _2$
MNIST on LeNet	2.33	<b>0.43</b>
MNIST on AlexNet	2.53	<b>1.15</b>
CIFAR-10 on ResNet-20	1.19	<b>1.11</b>

Table 3 reports the accuracy of the learned baseline values on the above functions. In most cases, the accuracy was above 90%, showing that our method could effectively learn correct baseline values. A few functions in (Tsang et al., 2018) did not have salient interaction patterns, which caused errors in the estimation of baseline values.

**Correctness of the computed Shapley values.** We further verified the correctness of the computed Shapley values on the extended Addition-Multiplication dataset (Zhang et al., 2021). We added the subtraction operation to avoid all baseline values being zero. Harsanyi (1963) provided us a new perspective to compute the Shapley value, *i.e.* the Shapley value was considered as a uniform assignment of attributions from each interaction pattern to its compositional variables. This enabled us to determine the ground-truth Shapley value of variables based on interactions without baseline values. For example, function  $f(\mathbf{x}) = 3x_1x_2 + 5x_3x_4 + x_5$  where  $\mathbf{x} = [1, 1, 1, 1, 1]$  contained three interaction patterns according to the principle of the most simplified interaction. Accordingly, the ground-truth Shapley values were  $\hat{\phi}_1 = \hat{\phi}_2 = 3/2$ ,  $\hat{\phi}_3 = \hat{\phi}_4 = 5/2$ , and  $\hat{\phi}_5 = 1$ . Please see Appendix E.7 for more details. We computed Shapley values of variables in the extended Addition-Multiplication dataset using different baseline values, and compared their accuracy in Table 4. The result shows that our method exhibited the highest accuracy.

**Recovering original samples from adversarial examples.** Let  $x$  denote the normal sample, and let  $x^{\text{adv}} = x + \delta$  denote the adversarial example generated by (Madry et al., 2018). According to (Ren et al., 2021), the adversarial example  $x^{\text{adv}}$  mainly created out-of-distribution bivariate interactions with high-order contexts, which were actually related to the high-order interactions in this paper. Thus, in the scenario of this study, the adversarial utility was owing to out-of-distribution high-order interactions. The removal of input variables was supposed to remove most high-order interactions.

Therefore, the baseline value can be considered as the recovery of the original sample. In this way, we used the adversarial example  $x^{\text{adv}}$  to initialize baseline values before learning, and used  $L_{\text{marginal}}$  to learn baseline values. If the learned baseline values  $\mathbf{b}$  satisfy  $\|\mathbf{b} - x\|_1 \leq \|x^{\text{adv}} - x\|_1$ , we considered that our method successfully recovered the original sample to some extent. We conducted experiments using LeNet (LeCun et al., 1998), AlexNet (Krizhevsky et al., 2012), and ResNet-20 (He et al., 2016) on the MNIST dataset (LeCun et al., 1998) ( $\|\delta\|_\infty \leq 32/255$ ) and the CIFAR-10 dataset (Krizhevsky et al., 2009) ( $\|\delta\|_\infty \leq 8/255$ ). Table 5 shows that our method recovered original samples from adversarial examples, which demonstrated the effectiveness of our method. Please see Appendix E.8 for more discussions.

## 5 CONCLUSIONS

In this paper, we have innovatively defined the absence state of input variables based on the multi-variate interaction patterns in game theory. Based on this, we have formulated optimal baseline values for the computation of the Shapley value. Then, we have proposed an approximate yet efficient method to learn optimal baseline values that represent the absence states of input variables. Experimental results have demonstrated the effectiveness of our method.

**Reproducibility Statement.** This research mainly focuses on both the theoretical formulation of baseline values and the learning of baseline values in game theory. Appendix B and Appendix D provide proofs for all theoretical results in the paper. For the learning of baseline values, we have discussed all experimental settings about datasets and models in the first paragraph of Section 4 and Appendix E, which ensure the reproducibility. Furthermore, we will release the code when the paper is accepted.

## REFERENCES

- Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *arXiv preprint arXiv:1903.10464*, 2019.
- Chirag Agarwal and Anh Nguyen. Explaining an image classifier’s decisions using generative models. *arXiv preprint arXiv:1910.04256*, 2019.
- Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pp. 272–281. PMLR, 2019.
- Arthur Asuncion and David Newman. Uci machine learning repository, 2007.
- Robert J Aumann and Lloyd S Shapley. *Values of non-atomic games*. Princeton University Press, 2015.
- Peer Bork, Lars J Jensen, Christian Von Mering, Arun K Ramani, Insuk Lee, and Edward M Marcotte. Protein interaction networks from yeast to human. *Current opinion in structural biology*, 14(3): 292–299, 2004.
- Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009.
- Ian Covert, Scott Lundberg, and Su-In Lee. Feature removal is a unifying principle for model explanation methods. *arXiv preprint arXiv:2011.03623*, 2020a.
- Ian Covert, Scott Lundberg, and Su-In Lee. Understanding global feature contributions through additive importance measures. *arXiv preprint arXiv:2004.00668*, 2020b.
- Tianyu Cui, Pekka Marttinen, and Samuel Kaski. Learning global pairwise interactions with bayesian neural networks. *arXiv preprint arXiv:1901.08361*, 2019.
- Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *arXiv preprint arXiv:1705.07857*, 2017.
- Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4829–4837, 2016.
- Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2950–2958, 2019.
- Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3429–3437, 2017.
- Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=OPyWRrcjVQw>.
- Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28(4):547–565, 1999.

- Ulrike Grömping. Estimators of relative importance in linear regression based on variance decomposition. *The American Statistician*, 61(2):139–147, 2007.
- Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018.
- John C Harsanyi. A simplified bargaining model for the n-person cooperative game. *International Economic Review*, 4(2):194–220, 1963.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *arXiv preprint arXiv:2002.04138*, 2020.
- Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *International Conference on Learning Representations*, 2019.
- Alon Keinan, Ben Sandbank, Claus C Hilgetag, Isaac Meilijson, and Eytan Ruppin. Fair attribution of functional contribution in artificial and biological networks. *Neural computation*, 16(9):1887–1915, 2004.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pp. 5491–5500. PMLR, 2020.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Richard Harold Lindeman. Introduction to bivariate and multivariate analysis. Technical report, 1980.
- Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>.
- Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.
- Luke Merrick and Ankur Taly. The explanation game: Explaining machine learning models using shapley values. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 17–38. Springer, 2020.
- Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks, 2015. URL <https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>.
- W James Murdoch, Peter J Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *International Conference on Learning Representations*, 2018.

- Jie Ren, Die Zhang, Yisen Wang, Lu Chen, Zhanpeng Zhou, Xu Cheng, Xin Wang, Yiting Chen, Jie Shi, and Quanshi Zhang. Game-theoretic understanding of adversarially learned features. *arXiv preprint arXiv:2103.07364*, 2021.
- Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.
- Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.
- Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- Chandan Singh, W James Murdoch, and Bin Yu. Hierarchical interpretations for neural network predictions. In *International Conference on Learning Representations*, 2018.
- Daria Sorokina, Rich Caruana, Mirek Riedewald, and Daniel Fink. Detecting statistical interactions with additive groves of trees. In *Proceedings of the 25th international conference on Machine learning*, pp. 1000–1007, 2008.
- Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.
- Erik Štrumbelj, Igor Kononenko, and M Robnik Šikonja. Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering*, 68(10):886–904, 2009.
- Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020.
- Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International Conference on Machine Learning*, pp. 9269–9278. PMLR, 2020.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3319–3328, 2017.
- Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *International Conference on Machine Learning*, pp. 9259–9268. PMLR, 2020.
- Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. In *International Conference on Learning Representations*, 2018.
- Robert J Weber. Probabilistic values for games. *The Shapley Value. Essays in Honor of Lloyd S. Shapley*, pp. 101–119, 1988.
- Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *International Conference on Machine Learning*, 2015.
- Hao Zhang, Yichen Xie, Longjie Zheng, Die Zhang, and Quanshi Zhang. Interpreting multivariate shapley interactions in dnns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10877–10886, 2021.
- Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations*, 2015.
- Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

## A MORE RELATED WORKS

**Shapley values.** The Shapley value (Shapley, 1953) was first proposed in game theory, which was considered as an unbiased distribution of the overall reward in a game to each player. Lindeman (1980) and Grömping (2007) used the Shapley value to attribute the correlation coefficient of a linear regression to input features. Štrumbelj et al. (2009); Štrumbelj & Kononenko (2014) used the Shapley value to attribute the prediction of a model to input features. Bork et al. (2004) used the Shapley value to measure importances of protein interactions in large, complex biological interaction networks. Keinan et al. (2004) employed the Shapley value to measure causal effects in neurophysiological models. Sundararajan et al. (2017) proposed Integrated Gradients based on the Aumann-Shapley (Aumann & Shapley, 2015) cost-sharing technique. Besides above local explanations, Covert et al. (2020b) focused on the global interpretability.

In order to compute the Shapley value in deep models efficiently, Lundberg & Lee (2017) proposed various approximations for Shapley value in DNNs. Lundberg et al. (2018) further computed the Shapley value on tree ensembles. Aas et al. (2019) generalized the approximation method in (Lundberg & Lee, 2017) to the case when features were related to each other. Ancona et al. (2019) further formulated a polynomial-time approximation of Shapley values for DNNs.

Unlike previous studies, we rethink and formulate baseline values from the perspective of game theory. We define the absence state of input variables based on the multi-variate interaction, and further propose a method to learn optimal baseline values.

**Interactions.** Interactions between input variables of deep models have been widely investigated in recent years. Sorokina et al. (2008) proposed an approach to detect interactions of input variables in an additive model. Tsang et al. (2018) measured interactions of weights in a DNN. Murdoch et al. (2018); Singh et al. (2018); Jin et al. (2019) used the contextual decomposition (CD) technique to extract variable interactions. Cui et al. (2019) proposed a non-parametric probabilistic method to measure interactions using a Bayesian neural network. In game theory, Grabisch & Roubens (1999); Lundberg et al. (2018) proposed and used the Shapley interaction index based on Shapley values. Janizek et al. (2020) extended the Integrated Gradients method (Sundararajan et al., 2017) to explain pairwise feature interactions in DNNs. Sundararajan et al. (2020) defined the Shapley-Taylor index to measure interactions over binary features. In this paper, we use the multi-variate interaction of input variables based on the Harsanyi dividend (Harsanyi, 1963). Our interaction metric has strong connections to (Grabisch & Roubens, 1999), but represents elementary interaction patterns in a more detailed manner, and satisfies the efficiency axiom.

## B EXTENDED DISCUSSIONS ABOUT THE INTERACTION BENEFIT BASED ON HARSANYI DIVIDEND

This section provides extended discussions about the interaction benefit based on the Harsanyi dividend (Harsanyi, 1963), and provides proofs of axioms that the Harsanyi dividend satisfies.

Given a trained model and the input sample  $x$  with  $n$  variables  $N = \{1, 2, \dots, n\}$ , let  $v(S)$  denote the model output when only variables in  $S$  are given. Then,  $v(N) - v(\emptyset)$  represents the overall inference benefit owing to all input variables in  $x$ , *w.r.t.* the model output without given any variables. In an input sample, different input variables interact with each other for inference, instead of working individually. The Harsanyi dividend  $I(S)$  measures the benefit to the model output from the interaction between variables in the subset  $S \subseteq N$ . Furthermore, the Harsanyi dividend ensures that the overall benefit  $v(N) - v(\emptyset)$  can be decomposed into the sum of  $I(S)$  of different subsets  $\{S | S \subseteq N, S \neq \emptyset\}$ .

$$v(N) - v(\emptyset) = \sum_{S \subseteq N, S \neq \emptyset} I(S) \quad (8)$$

For example, let us consider the  $S_1 = \{\text{head}\}$  denote the bird head pattern in Figure 6. Patches inside  $S_1$  collaborate with each other to form the head pattern, and to contribute an interaction benefit  $I(S_1)$  for the model output. Then, the overall benefit of all patches in the image  $v(N) - v(\emptyset)$  can be represented as the sum of various patterns, such as the head pattern  $S_1$  with its benefit  $I(S_1)$ , the tail pattern  $S_2$  with the benefit  $I(S_2)$ , the torso pattern  $S_3$  with the utility  $I(S_3)$ , etc.

The Harsanyi dividend  $I(S)$  is defined to measure the additional benefit from the collaboration of input variables in  $S$ , in comparison with the benefit when they work individually or form smaller


$$v(N) = I(S_1) + I(S_2) + I(S_3) + I(S_4) + \dots + v(\emptyset)$$


Figure 6: Sketch for the multi-variate interaction. The model output of a DNN can be decomposed into interaction benefits  $I(S)$  of different patterns.

patterns. Specifically, let  $v(S) - v(\emptyset)$  denote the overall benefit from all variables in  $S$ , then we remove the marginal benefits owing to collaborations of all smaller subsets  $L$  of variables in  $S$ , *i.e.*  $\{I(L) | L \subsetneq S, L \neq \emptyset\}$ .

$$I(S) \stackrel{\text{def}}{=} \underbrace{v(S) - v(\emptyset)}_{\text{the benefit from all variables in } S} - \sum_{L \subsetneq S, L \neq \emptyset} I(L) \quad (9)$$

### B.1 PROOF OF AXIOMS OF THE INTERACTION BENEFIT BASED ON HARSANYI DIVIDEND

In Section 3.2 of the paper, we claim that we extend the linearity, dummy, symmetry axioms of Shapley values to the interaction benefit based on Harsanyi dividend. Here, we provide details and proofs for these axioms.

**(1) Linearity property (axiom):** If we merge outputs of two models,  $u(S) = w(S) + v(S)$ , then,  $\forall S \subseteq N$ , the interaction  $I_u(S)$  *w.r.t.* the new output  $u$  can be decomposed into  $I_u(S) = I_w(S) + I_v(S)$ .

• *Proof:*

$$\begin{aligned} I_u(S) &= \sum_{L \subseteq S} (-1)^{|S|-|L|} u(L) \\ &= \sum_{L \subseteq S} (-1)^{|S|-|L|} [w(L) + v(L)] \\ &= \sum_{L \subseteq S} (-1)^{|S|-|L|} w(L) + \sum_{L \subseteq S} (-1)^{|S|-|L|} v(L) \\ &= I_w(S) + I_v(S) \end{aligned}$$

**(2) Dummy property:** The dummy variable  $i \in N$  satisfies  $\forall S \subseteq N \setminus \{i\}$ ,  $v(S \cup \{i\}) = v(S) + v(\{i\})$ . It means that the variable  $i$  has no interactions with other variables, *i.e.*  $\forall S \subseteq N \setminus \{i\}$ ,  $I(S \cup \{i\}) = 0$ .

• *Proof:*

$$\begin{aligned} I(S \cup \{i\}) &= \sum_{L \subseteq S \cup \{i\}} (-1)^{|S \cup \{i\}|-|L|} v(L) \\ &= \sum_{L \subseteq S} (-1)^{|S \cup \{i\}|-|L|} v(L) + \sum_{L \subseteq S} (-1)^{|S \cup \{i\}|-|L|} v(L \cup \{i\}) \\ &= \sum_{L \subseteq S} (-1)^{|S \cup \{i\}|-|L|} v(L) + \sum_{L \subseteq S} (-1)^{|S \cup \{i\}|-|L|} [v(L) + v(\{i\})] \\ &= \sum_{L \subseteq S} (-1)^{|S \cup \{i\}|-|L|} [-v(L) + v(L)] + \sum_{L \subseteq S} (-1)^{|S \cup \{i\}|-|L|} v(\{i\}) \\ &= \left( \sum_{L \subseteq S} (-1)^{|S \cup \{i\}|-|L|} \right) v(\{i\}) \\ &= [1 + (-1)^{|S \cup \{i\}|-|S|}] v(\{i\}) \\ &= 0 \end{aligned}$$

**(3) Symmetry property:** If input variables  $i, j \in N$  have same cooperations with other variables  $\forall S \subseteq N \setminus \{i, j\}, v(S \cup \{i\}) = v(S \cup \{j\})$ , then they have same interactions with other variables,  $\forall S \subseteq N \setminus \{i, j\}, I(S \cup \{i\}) = I(S \cup \{j\})$ .

• *Proof:*

$$\begin{aligned}
I(S \cup \{i\}) &= \sum_{L \subseteq S \cup \{i\}} (-1)^{|S|-|L|} v(L) \\
&= \sum_{L \subseteq S} (-1)^{|S|+1-|L|} v(L) + \sum_{L \subseteq S} (-1)^{|S|-|L|} v(L \cup \{i\}) \\
&= \sum_{L \subseteq S} (-1)^{|S|+1-|L|} v(L) + \sum_{L \subseteq S} (-1)^{|S|-|L|} v(L \cup \{j\}) \\
&= \sum_{L \subseteq S \cup \{j\}} (-1)^{|S|-|L|} v(L) \\
&= I(S \cup \{j\})
\end{aligned}$$

**(4) Efficiency property, proved by Harsanyi (1963):** The output of a model can be decomposed into interactions of different subsets of variables,  $v(N) = v(\emptyset) + \sum_{S \subseteq N, S \neq \emptyset} I(S)$ .

• *Proof:*

$$\begin{aligned}
\text{right} &= v(\emptyset) + \sum_{S \subseteq N, S \neq \emptyset} I(S) \\
&= v(\emptyset) + \sum_{S \subseteq N, S \neq \emptyset} \sum_{L \subseteq S} (-1)^{|S|-|L|} v(L) \\
&= \sum_{S \subseteq N} \sum_{L \subseteq S} (-1)^{|S|-|L|} v(L) \\
&= \sum_{L \subseteq N} \sum_{K \subseteq N \setminus L} (-1)^{|K|} v(L) \quad \% \text{ Let } K = S \setminus L \\
&= \sum_{L \subseteq N} \left[ \sum_{|K|=0}^{n-|L|} \binom{n-|L|}{|K|} (-1)^{|K|} \right] v(L) \\
&= \sum_{L \subseteq N} \left[ (1 + (-1))^{n-|L|} \right] v(L) \\
&= v(N) = \text{left}
\end{aligned}$$

## B.2 PROOF OF THE CONNECTION TO SHAPLEY VALUES

Let  $\phi(i)$  denote the Shapley value (Shapley, 1953) of an input variable  $i$ . Then, its Shapley value can be represented as the weighted sum of the Harsanyi dividend,  $\phi(i) = \sum_{S \subseteq N \setminus \{i\}} \frac{1}{|S|+1} I(S \cup \{i\})$ . This connection has been proved in (Harsanyi, 1963).

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{1}{|S|+1} I(S \cup \{i\}) \tag{10}$$

• *Proof:*

$$\begin{aligned}
\text{right} &= \sum_{S \subseteq N \setminus \{i\}} \frac{1}{|S|+1} I(S \cup \{i\}) \\
&= \sum_{S \subseteq N \setminus \{i\}} \frac{1}{|S|+1} \left[ \sum_{L \subseteq S} (-1)^{|S|+1-|L|} v(L) + \sum_{L \subseteq S} (-1)^{|S|-|L|} v(L \cup \{i\}) \right]
\end{aligned}$$

$$\begin{aligned}
&= \sum_{S \subseteq N \setminus \{i\}} \frac{1}{|S|+1} \sum_{L \subseteq S} (-1)^{|S|-|L|} [v(L \cup \{i\}) - v(L)] \\
&= \sum_{L \subseteq N \setminus \{i\}} \sum_{K \subseteq N \setminus L \setminus \{i\}} \frac{(-1)^{|K|}}{|K|+|L|+1} [v(L \cup \{i\}) - v(L)] \quad \% \text{ Let } K = S \setminus L \\
&= \sum_{L \subseteq N \setminus \{i\}} \left( \sum_{k=0}^{n-1-|L|} \frac{(-1)^k}{k+|L|+1} \binom{n-1-|L|}{k} \right) [v(L \cup \{i\}) - v(L)] \quad \% \text{ Let } k = |K| \\
&= \sum_{L \subseteq N \setminus \{i\}} \frac{|L|!(n-1-|L|)!}{n!} [v(L \cup \{i\}) - v(L)] \quad \% \text{ by the property of combinatorial number} \\
&= \phi_i = \text{left}
\end{aligned}$$

### B.3 PROOF OF CONNECTIONS BETWEEN THE HARSANYI DIVIDEND AND OTHER GAME-THEORETIC INTERACTION METRICS

In this section, we prove the relationship between the Harsanyi dividend and other game-theoretic interaction metrics, including the Shapley interaction index (Grabisch & Roubens, 1999) and the Shapley Taylor interaction index (Sundararajan et al., 2020).

**Lemma (Connection to the marginal benefit).**  $\Delta v_T(S) = \sum_{L \subseteq T} (-1)^{|T|-|L|} v(L \cup S)$  denotes the marginal benefit (Grabisch & Roubens, 1999) of variables in  $T \subseteq N \setminus S$  given the environment  $S$ . We have proven that  $\Delta v_T(S)$  can be decomposed into the sum of interaction benefits of  $T$  and sub-environments of  $S$ , i.e.  $\Delta v_T(S) = \sum_{S' \subseteq S} I(T \cup S')$ .

• *Proof:* By the definition of the marginal benefit, we have

$$\begin{aligned}
\Delta v_T(S) &= \sum_{L \subseteq T} (-1)^{|T|-|L|} v(L \cup S) \\
&= \sum_{L \subseteq T} (-1)^{|T|-|L|} \sum_{K \subseteq L \cup S} I(K) \\
&= \sum_{L \subseteq T} (-1)^{|T|-|L|} \sum_{L' \subseteq L} \sum_{S' \subseteq S} I(L' \cup S') \quad // \text{ since } L \cap S = \emptyset \\
&= \sum_{S' \subseteq S} \left[ \sum_{L \subseteq T} (-1)^{|T|-|L|} \sum_{L' \subseteq L} I(L' \cup S') \right] \\
&= \sum_{S' \subseteq S} \left[ \sum_{L' \subseteq T} \sum_{\substack{L \subseteq T \\ L \supseteq L'}} (-1)^{|T|-|L|} I(L' \cup S') \right] \\
&= \sum_{S' \subseteq S} \left[ \underbrace{I(S' \cup T)}_{L'=T} + \sum_{L' \subsetneq T} \left( \sum_{l=|L'|}^{|T|} \binom{|T|-|L'|}{l-|L'|} (-1)^{|T|-|L|} I(L' \cup S') \right) \right]_{L' \subsetneq T} \\
&= \sum_{S' \subseteq S} \left[ I(S' \cup T) + \sum_{L' \subsetneq T} \left( I(L' \cup S') \cdot \underbrace{\sum_{l=|L'|}^{|T|} \binom{|T|-|L'|}{l-|L'|} (-1)^{|T|-|L|}}_{=0} \right) \right] \\
&= \sum_{S' \subseteq S} I(S' \cup T)
\end{aligned}$$

**Theorem 1 (Connection to the Shapley interaction index)** Given a subset of input variables  $T \subseteq N$ ,  $I^{\text{Shapley}}(T) = \sum_{S \subseteq N \setminus T} \frac{|S|!(|N|-|S|-|T|)!}{(|N|-|T|+1)!} \Delta v_T(S)$  denotes the Shapley interaction in-



dex (Grabisch & Roubens, 1999) of  $T$ . We have proven that the Shapley interaction index can be represented as the weighted sum of utilities of interaction patterns,  $I^{\text{Shapley}}(T) = \sum_{S \subseteq N \setminus T} \frac{1}{|S|+1} I(S \cup T)$ .

• *Proof:*

$$\begin{aligned}
I^{\text{Shapley}}(T) &= \sum_{S \subseteq N \setminus T} \frac{|S|!(|N| - |S| - |T|)!}{(|N| - |T| + 1)!} \Delta v_T(S) \\
&= \frac{1}{|N| - |T| + 1} \sum_{m=0}^{|N|-|T|} \frac{1}{\binom{|N|-|T|}{m}} \sum_{\substack{S \subseteq N \setminus T \\ |S|=m}} \Delta v_T(S) \\
&= \frac{1}{|N| - |T| + 1} \sum_{m=0}^{|N|-|T|} \frac{1}{\binom{|N|-|T|}{m}} \sum_{\substack{S \subseteq N \setminus T \\ |S|=m}} \left[ \sum_{L \subseteq S} I(L \cup T) \right] \\
&= \frac{1}{|N| - |T| + 1} \sum_{L \subseteq N \setminus T} \sum_{m=|L|}^{|N|-|T|} \frac{1}{\binom{|N|-|T|}{m}} \sum_{\substack{S \subseteq N \setminus T \\ |S|=m \\ S \supseteq L}} I(L \cup T) \\
&= \frac{1}{|N| - |T| + 1} \sum_{L \subseteq N \setminus T} \sum_{m=|L|}^{|N|-|T|} \frac{1}{\binom{|N|-|T|}{m}} \binom{|N| - |L| - |T|}{m - |L|} I(L \cup T) \\
&= \frac{1}{|N| - |T| + 1} \sum_{L \subseteq N \setminus T} I(L \cup T) \underbrace{\sum_{k=0}^{|N|-|L|-|T|} \frac{1}{\binom{|N|-|T|}{|L|+k}} \binom{|N| - |L| - |T|}{k}}_{w_L}
\end{aligned}$$

Then, we leverage the following properties of combinatorial numbers and the Beta function to simplify the term  $w_L = \sum_{k=0}^{|N|-|L|-|T|} \frac{1}{\binom{|N|-|T|}{|L|+k}} \cdot \binom{|N|-|L|-|T|}{k}$ .

(i) *A property of combinatorial numbers.*  $m \cdot \binom{n}{m} = n \cdot \binom{n-1}{m-1}$ .

(ii) *The definition of the Beta function.* For  $p, q > 0$ , the Beta function is defined as  $B(p, q) = \int_0^1 x^{p-1} (1-x)^{q-1} dx$ .

(iii) *Connections between combinatorial numbers and the Beta function.*

- When  $p, q \in \mathbb{Z}^+$ , we have  $B(p, q) = \frac{1}{q \cdot \binom{p+q-1}{p-1}}$ .
- For  $m, n \in \mathbb{Z}^+$  and  $n > m$ , we have  $\binom{n}{m} = \frac{1}{m \cdot B(n-m+1, m)}$ .

Hence, we leverage the properties of combinatorial numbers and the Beta function to simplify  $w_L$ .

$$\begin{aligned}
w_L &= \sum_{k=0}^{|N|-|L|-|T|} \frac{1}{\binom{|N|-|T|}{|L|+k}} \binom{|N| - |L| - |T|}{k} \\
&= \sum_{k=0}^{|N|-|L|-|T|} \binom{|N| - |L| - |T|}{k} \cdot (|L| + k) \cdot B(|N| - |L| - |T| - k + 1, |L| + k) \\
&= \sum_{k=0}^{|N|-|L|-|T|} |L| \cdot \binom{|N| - |L| - |T|}{k} \cdot B(|N| - |L| - |T| - k + 1, |L| + k) \quad \dots \textcircled{1} \\
&\quad + \sum_{k=0}^{|N|-|L|-|T|} k \cdot \binom{|N| - |L| - |T|}{k} \cdot B(|N| - |L| - |T| - k + 1, |L| + k) \quad \dots \textcircled{2}
\end{aligned}$$

Then, we solve ① and ② respectively. For ①, we have

$$\begin{aligned}
\textcircled{1} &= \int_0^1 |L| \sum_{k=0}^{|N|-|L|-|T|} \binom{|N|-|L|-|T|}{k} \cdot x^{|N|-|L|-|T|-k} \cdot (1-x)^{|L|+k-1} dx \\
&= \int_0^1 |L| \cdot \underbrace{\left[ \sum_{k=0}^{|N|-|L|-|T|} \binom{|N|-|L|-|T|}{k} \cdot x^{|N|-|L|-|T|-k} \cdot (1-x)^k \right]}_{=1} \cdot (1-x)^{|L|-1} dx \\
&= \int_0^1 |L| \cdot (1-x)^{|L|-1} dx = 1
\end{aligned}$$

For  $\textcircled{2}$ , we have

$$\begin{aligned}
\textcircled{2} &= \sum_{k=1}^{|N|-|L|-|T|} (|N|-|L|-|T|) \binom{|N|-|L|-|T|-1}{k-1} \cdot B(|N|-|L|-|T|-k+1, |L|+k) \\
&= (|N|-|L|-|T|) \sum_{k'=0}^{|N|-|L|-|T|-1} \binom{|N|-|L|-|T|-1}{k'} \cdot B(|N|-|L|-|T|-k', |L|+k'+1) \\
&= (|N|-|L|-|T|) \int_0^1 \sum_{k'=0}^{|N|-|L|-|T|-1} \binom{|N|-|L|-|T|-1}{k'} \cdot x^{|N|-|L|-|T|-k'-1} \cdot (1-x)^{|L|+k'} dx \\
&= (|N|-|L|-|T|) \int_0^1 \underbrace{\left[ \sum_{k'=0}^{|N|-|L|-|T|-1} \binom{|N|-|L|-|T|-1}{k'} \cdot x^{|N|-|L|-|T|-k'-1} \cdot (1-x)^{k'} \right]}_{=1} \cdot (1-x)^{|L|} dx \\
&= (|N|-|L|-|T|) \int_0^1 (1-x)^{|L|} dx = \frac{|N|-|L|-|T|}{|L|+1}
\end{aligned}$$

Hence, we have

$$w_L = \textcircled{1} + \textcircled{2} = 1 + \frac{|N|-|L|-|T|}{|L|+1} = \frac{|N|-|T|+1}{|L|+1}$$

Therefore, we proved that  $I^{\text{Shapley}}(T) = \frac{1}{|N|-|T|+1} \sum_{L \subseteq N \setminus T} w_L \cdot I(L \cup T) = \sum_{L \subseteq N \setminus T} \frac{1}{|L|+1} I(L \cup T)$ .

**Theorem 2 (Connection to the Shapley Taylor interaction index)** Given a subset of input variables  $T \subseteq N$ , let  $I^{\text{Shapley-Taylor}}(T)$  denote the Shapley Taylor interaction index (Sundararajan et al., 2020) of order  $k$  for  $T$ . We have proven that the Shapley Taylor interaction index can be represented as the weighted sum of interaction utilities, *i.e.*  $I^{\text{Shapley-Taylor}}(T) = I(T)$  if  $|T| < k$ ;  $I^{\text{Shapley-Taylor}}(T) = \sum_{S \subseteq N \setminus T} \binom{|S|+k}{k}^{-1} I(S \cup T)$  if  $|T| = k$ ; and  $I^{\text{Shapley-Taylor}}(T) = 0$  if  $|T| > k$ .

• *Proof:* By the definition of the Shapley Taylor interaction index,

$$I^{\text{Shapley-Taylor}(k)}(T) = \begin{cases} \Delta v_T(\emptyset) & \text{if } |T| < k \\ \frac{k}{|N|} \sum_{S \subseteq N \setminus T} \frac{1}{\binom{|N|-1}{|S|}} \Delta v_T(S) & \text{if } |T| = k \\ 0 & \text{if } |T| > k \end{cases}$$

When  $|T| < k$ , by the definition of the interaction utility in Equation (3), we have

$$I^{\text{Shapley-Taylor}(k)}(T) = \Delta v_T(\emptyset) = \sum_{L \subseteq T} (-1)^{|T|-|L|} \cdot v(L) = I(T).$$

When  $|T| = k$ , we have

$$\begin{aligned}
I^{\text{Shapley-Taylor}(k)}(T) &= \frac{k}{|N|} \sum_{S \subseteq N \setminus T} \frac{1}{\binom{|N|-1}{|S|}} \cdot \Delta v_T(S) \\
&= \frac{k}{|N|} \sum_{m=0}^{|N|-k} \sum_{\substack{S \subseteq N \setminus T \\ |S|=m}} \frac{1}{\binom{|N|-1}{|S|}} \cdot \Delta v_T(S) \\
&= \frac{k}{|N|} \sum_{m=0}^{|N|-k} \sum_{\substack{S \subseteq N \setminus T \\ |S|=m}} \frac{1}{\binom{|N|-1}{|S|}} \left[ \sum_{L \subseteq S} I(L \cup T) \right] \\
&= \frac{k}{|N|} \sum_{L \subseteq N \setminus T} \sum_{m=|L|}^{|N|-k} \frac{1}{\binom{|N|-1}{|S|}} \sum_{\substack{S \subseteq N \setminus T \\ |S|=m \\ S \supseteq L}} I(L \cup T) \\
&= \frac{k}{|N|} \sum_{L \subseteq N \setminus T} \sum_{m=|L|}^{|N|-k} \frac{1}{\binom{|N|-1}{|S|}} \binom{|N|-|L|-k}{m-|L|} I(L \cup T) \\
&= \frac{k}{|N|} \sum_{L \subseteq N \setminus T} I(L \cup T) \underbrace{\sum_{m=0}^{|N|-|L|-k} \frac{1}{\binom{|N|-1}{|L|+m}} \binom{|N|-|L|-k}{m}}_{w_L}
\end{aligned}$$

Just like the proof of Theorem 1, we leverage the properties of combinatorial numbers and the Beta function to simplify  $w_L$ .

$$\begin{aligned}
w_L &= \sum_{m=0}^{|N|-|L|-k} \frac{1}{\binom{|N|-1}{|L|+m}} \binom{|N|-|L|-k}{m} \\
&= \sum_{m=0}^{|N|-|L|-k} \binom{|N|-|L|-k}{m} \cdot (|L|+m) \cdot B(|N|-|L|-m, |L|+m) \\
&= \sum_{m=0}^{|N|-|L|-k} |L| \cdot \binom{|N|-|L|-k}{m} \cdot B(|N|-|L|-m, |L|+m) \quad \dots \textcircled{1} \\
&\quad + \sum_{m=0}^{|N|-|L|-k} m \cdot \binom{|N|-|L|-k}{m} \cdot B(|N|-|L|-m, |L|+m) \quad \dots \textcircled{2}
\end{aligned}$$

Then, we solve  $\textcircled{1}$  and  $\textcircled{2}$  respectively. For  $\textcircled{1}$ , we have

$$\begin{aligned}
\textcircled{1} &= \int_0^1 |L| \cdot \sum_{m=0}^{|N|-|L|-k} \binom{|N|-|L|-k}{m} \cdot x^{|N|-|L|-m-1} \cdot (1-x)^{|L|+m-1} dx \\
&= \int_0^1 |L| \cdot \underbrace{\left[ \sum_{m=0}^{|N|-|L|-k} \binom{|N|-|L|-k}{m} \cdot x^{|N|-|L|-m-k} \cdot (1-x)^m \right]}_{=1} \cdot x^{k-1} \cdot (1-x)^{|L|-1} dx \\
&= \int_0^1 |L| \cdot x^{k-1} \cdot (1-x)^{|L|-1} dx = |L| \cdot B(k, |L|) = \frac{1}{\binom{|L|+k-1}{k-1}}
\end{aligned}$$

For  $\textcircled{2}$ , we have

$$\begin{aligned}
\textcircled{2} &= \sum_{m=1}^{|N|-|L|-k} (|N|-|L|-k) \cdot \binom{|N|-|L|-k-1}{m-1} \cdot B(|N|-|L|-m, |L|+m) \\
&= \sum_{m'=0}^{|N|-|L|-k-1} (|N|-|L|-k) \cdot \binom{|N|-|L|-k-1}{m'} \cdot B(|N|-|L|-m'-1, |L|+m'+1) \\
&= \int_0^1 (|N|-|L|-k) \sum_{m'=0}^{|N|-|L|-k-1} \binom{|N|-|L|-k-1}{m'} \cdot x^{|N|-|L|-m'-2} \cdot (1-x)^{|L|+m'} dx \\
&= \int_0^1 (|N|-|L|-k) \underbrace{\left[ \sum_{m'=0}^{|N|-|L|-k-1} \binom{|N|-|L|-k-1}{m'} \cdot x^{|N|-|L|-m'-k-1} \cdot (1-x)^{m'} \right]}_{=1} \cdot x^{k-1} \cdot (1-x)^{|L|} dx \\
&= \int_0^1 (|N|-|L|-k) \cdot x^{k-1} \cdot (1-x)^{|L|} dx = (|N|-|L|-k) \cdot B(k, |L|+1) \\
&= \frac{|N|-|L|-k}{(|L|+1) \binom{|L|+k}{k-1}}
\end{aligned}$$

Hence, we have

$$\begin{aligned}
w_L &= \textcircled{1} + \textcircled{2} = \frac{1}{\binom{|L|+k-1}{k-1}} + \frac{|N|-|L|-k}{(|L|+1) \binom{|L|+k}{k-1}} \\
&= \frac{|L|! \cdot (k-1)!}{(|L|+k-1)!} + \frac{|N|-|L|-k}{|L|+1} \cdot \frac{(|L|+1)! \cdot (k-1)!}{(|L|+k)!} \\
&= \frac{|L|! \cdot (k-1)!}{(|L|+k-1)!} + \frac{|N|-|L|-k}{|L|+k} \cdot \frac{|L|! \cdot (k-1)!}{(|L|+k-1)!} \\
&= \left[ 1 + \frac{|N|-|L|-k}{|L|+k} \right] \cdot \frac{|L|! \cdot (k-1)!}{(|L|+k-1)!} \\
&= \frac{|N|}{|L|+k} \cdot \frac{|L|! \cdot (k-1)!}{(|L|+k-1)!} \\
&= \frac{|N|}{k} \cdot \frac{|L|! \cdot k!}{(|L|+k)!} \\
&= \frac{|N|}{k} \cdot \frac{1}{\binom{|L|+k}{k}}
\end{aligned}$$

Therefore, we proved that when  $|T| = k$ ,  $I^{\text{Shapley-Taylor}}(T) = \frac{k}{|N|} \sum_{L \subseteq N \setminus T} w_L \cdot I(L \cup T) = \frac{k}{|N|} \sum_{L \subseteq N \setminus T} \frac{|N|}{k} \cdot \frac{1}{\binom{|L|+k}{k}} \cdot I(L \cup T) = \sum_{L \subseteq N \setminus T} \binom{|L|+k}{k}^{-1} I(L \cup T)$ .

## C DISCUSSION ABOUT BASELINE VALUES BASED ON THE CONDITIONAL DISTRIBUTION AND THE MARGINAL DISTRIBUTIONS

This section provides more discussions about the baseline values based on the conditional distribution and the marginal distribution.

- **Baseline values based on the conditional distribution.** Instead of fixing baseline values as constants, some studies use varying baseline values, which are determined temporarily by the context  $S$  in a specific sample  $x$ , to compute  $v(S|x)$  given  $x$ . Some methods (Covert et al., 2020b; Frye et al., 2021) define  $v(S|x)$  by modeling the conditional distribution of variable values in  $\bar{S} = N \setminus S$  given the context  $S$ , i.e.  $v(S|x) = \mathbb{E}_{p(x'_{\bar{S}}|x_S)}[f(x_S \cup x'_{\bar{S}})]$ . However, these methods apply varying baseline values based on the dependency between input variables, which do not faithfully satisfy the linearity axiom and the nullity axiom of the Shapley value from some aspects (Sundararajan & Najmi, 2020).

Moreover, these methods compute a specific conditional distribution  $p(x'|x_S)$  for each of  $2^n$  contexts  $S$  with a very high computational cost.

*The nullity axiom.* The nullity axiom of the Shapley value ensures that for a null player  $i$  s.t.  $\forall S \subseteq N \setminus \{i\}, v(S \cup \{i\}) = v(S)$ , its Shapley value is  $\phi_i = 0$ . This axiom means that if the model is insensitive to an input variable, then the variable should have zero attributions. It seems that baseline values based on the conditional distribution satisfy this axiom mathematically, because  $\phi_i = \mathbb{E}_{S \subseteq N \setminus \{i\}} [v(S \cup \{i\}) - v(S)] = 0$ .

However, as discussed in (Sundararajan & Najmi, 2020), the conditional baseline does not faithfully satisfy the natural meaning of the nullity axiom. Let us consider an example where the model  $f(x) = x_1 + x_3$  and each input sample  $x = [x_1, x_2, x_3]$  contains three variables ( $N = \{1, 2, 3\}$ ). Each input variable  $x_i \in \{0, 1\}$  is a binary variable. In this case, because the variable  $x_2$  is not referenced in the model, the Shapley value of  $x_2$  is supposed to be  $\hat{\phi}_2 = 0$ . We assume that  $x_1 \sim \text{Bernoulli}(0.5)$ ,  $x_3 \sim \text{Bernoulli}(0.5)$ , and  $x_2$  is totally determined by  $x_3$ , i.e.  $x_2 = x_3$  in all samples. Based on such distribution of input variables, for the input sample  $x = [1, 1, 1]$ , we have

$$\begin{aligned} v(\emptyset) &= 0.25f([x'_1 = 0, x'_2 = 0, x'_3 = 0]) + 0.25f([x'_1 = 0, x'_2 = 1, x'_3 = 1]) \\ &\quad + 0.25f([x'_1 = 1, x'_2 = 0, x'_3 = 0]) + 0.25f([x'_1 = 1, x'_2 = 1, x'_3 = 1]) = 1 \\ v(\{1\}) &= 0.5f([x_1 = 1, x'_2 = 0, x'_3 = 0]) + 0.5f([x_1 = 1, x'_2 = 1, x'_3 = 1]) = 1.5 \\ v(\{2\}) &= 0.5f([x'_1 = 0, x_2 = 1, x'_3 = 1]) + 0.5f([x'_1 = 1, x_2 = 1, x'_3 = 1]) = 1.5 \\ v(\{3\}) &= 0.5f([x'_1 = 0, x'_2 = 1, x_3 = 1]) + 0.5f([x'_1 = 1, x'_2 = 1, x_3 = 1]) = 1.5 \\ v(\{1, 2\}) &= f([x_1 = 1, x_2 = 1, x'_3 = x_2 = 1]) = 2 \\ v(\{1, 3\}) &= f([x_1 = 1, x'_2 = x_3 = 1, x_3 = 1]) = 2 \\ v(\{2, 3\}) &= 0.5f([x'_1 = 0, x_2 = 1, x_3 = 1]) + 0.5f([x'_1 = 1, x_2 = 1, x_3 = 1]) = 1.5 \\ v(\{1, 2, 3\}) &= f([x_1 = 1, x_2 = 1, x_3 = 1]) = 2 \end{aligned}$$

Thus, the Shapley value of the variable  $x_2$  is

$$\begin{aligned} \phi_2 &= \frac{1}{3}[v(\{2\}) - v(\emptyset)] + \frac{1}{6}[v(\{1, 2\}) - v(\{1\}) + v(\{2, 3\}) - v(\{3\})] + \frac{1}{3}[v(\{1, 2, 3\}) - v(\{1, 3\})] \\ &= \frac{1}{3} \cdot \frac{1}{2} + \frac{1}{6} \cdot (0.5 + 0) + \frac{1}{3} \cdot 0 \\ &= \frac{1}{4} \neq 0 \end{aligned}$$

Obviously, the computed Shapley value of the variables  $x_2$  based on the conditional baseline is not 0, which contradicts with the fact that the model does not use the variable  $x_2$  in computation. This phenomenon has also been discussed in (Sundararajan & Najmi, 2020).

*The linearity axiom.* The linearity axiom of the Shapley value means that when we merge two models  $v, w$  into a new model  $u$ , then the Shapley values of variables in the two old models can also be merged. The baseline value based on the conditional distribution does not naturally satisfy this axiom. It is because the conditional distribution of input variables may be different in the two old models. There is an extremeness that some input variables in the old model  $v$  do not exist in the other old model  $w$ . In this case, the conditional distribution of variables in the three models  $u, v, w$  will be dramatically different. Please refer to (Sundararajan et al., 2017) for more detailed discussions.

*The problem with the computational cost.* Besides the above problem with axioms, the conditional baseline also suffers from the large computational cost, and is not suitable for continuous variables. In (Covert et al., 2020b),  $v(S|x) = \mathbb{E}_{p(x'|x_S)} [f(x_S \sqcup x'_S)]$  is approximately computed as  $v(S|x) = \mathbb{E}_{x' \in \Omega: x'_S = x_S} [f(x_S \sqcup x'_S)]$ , where  $\Omega$  denotes the set of all samples in the dataset. Unfortunately, for high-dimensional inputs with continuous variables, it is practically impossible to find inputs  $x'$  where  $x'_S = x_S$ . Frye et al. (2021) propose to use generative models to learn the distribution  $p(x'|x_S)$  for a specific context  $S$ . However, there are  $2^n$  different contexts  $S$ , leading to a very high computational cost.

• **Baseline values based on the marginal distribution.** Based on the assumption that input variables are independent with each other, Lundberg & Lee (2017) simplify the above conditional baseline to the marginal baseline, i.e.  $v(S|x) = \mathbb{E}_{p(x')} [f(x_S \sqcup x'_S)]$ . First, such assumption of variable

independence is not true in real applications, thereby hurting the trustworthiness of the computed Shapley values. Second, the real distribution of input samples  $p(x)$  is unknown. Thus,  $v(S|x)$  is approximated by  $\mathbb{E}_{x' \in \Omega}[f(x_S \sqcup x'_S)]$  in (Lundberg & Lee, 2017). In this case, the computational cost is also very large because we need to enumerate all samples in the dataset.

## D MULTI-ORDER SHAPLEY VALUES AND MARGINAL BENEFITS

In Section 3.3 of the paper, we claim that the Shapley value  $\phi_i$  can be decomposed into the sum of Shapley values of different orders  $\phi_i^{(m)}$ , and the sum of marginal benefits of different orders  $\Delta v_i(S)$ . Furthermore, the multi-order Shapley values and marginal benefits can be re-written as the sum of interaction benefits. This section provides proofs for the above claims.

First, we have proven the following decomposition of the Shapley value.

$$\phi_i = \frac{1}{n} \sum_{m=0}^{n-1} \phi_i^{(m)} = \frac{1}{n} \sum_{m=0}^{n-1} \mathbb{E}_{S \subseteq N \setminus \{i\}, |S|=m} \Delta v_i(S) \quad (11)$$

where the Shapley value of  $m$ -order  $\phi_i^{(m)} \stackrel{\text{def}}{=} \mathbb{E}_{S \subseteq N \setminus \{i\}, |S|=m} [v(S \cup \{i\}) - v(S)]$ , and the marginal benefit  $\Delta v_i(S) \stackrel{\text{def}}{=} v(S \cup \{i\}) - v(S)$ .

• *Proof:*

$$\begin{aligned} \phi_i &= \sum_{S \subseteq N} \frac{|S|!(n-1-|S|!)}{n!} [v(S \cup \{i\}) - v(S)] \\ &= \sum_{m=0}^{n-1} \sum_{S \subseteq N, |S|=m} \frac{|S|!(n-1-|S|!)}{n!} [v(S \cup \{i\}) - v(S)] \\ &= \frac{1}{n} \sum_{m=0}^{n-1} \sum_{S \subseteq N, |S|=m} \frac{|S|!(n-1-|S|!)}{(n-1)!} [v(S \cup \{i\}) - v(S)] \\ &= \frac{1}{n} \sum_{m=0}^{n-1} \mathbb{E}_{S \subseteq N, |S|=m} [v(S \cup \{i\}) - v(S)] \\ &= \frac{1}{n} \sum_{m=0}^{n-1} \phi_i^{(m)} \\ &= \frac{1}{n} \sum_{m=0}^{n-1} \mathbb{E}_{S \subseteq N \setminus \{i\}, |S|=m} \Delta v_i(S) \end{aligned}$$

**Connection between multi-variate interactions and multi-order marginal benefits.** Equation (6) in the main paper shows that the  $m$ -order marginal benefit can be decomposed as the sum of multi-variate interaction benefits. In the supplementary material, this section provides the proof for such decomposition.

$$\Delta v_i(S) = \sum_{L \subseteq S} I(L \cup \{i\}) \quad (12)$$

• *Proof:*

$$\begin{aligned} \text{right} &= \sum_{L \subseteq S} I(L \cup \{i\}) \\ &= \sum_{L \subseteq S} \left[ \sum_{K \subseteq L} (-1)^{|L|+1-|K|} v(K) + \sum_{K \subseteq L} (-1)^{|L|-|K|} v(K \cup \{i\}) \right] \\ &= \sum_{L \subseteq S} \sum_{K \subseteq L} (-1)^{|L|-|K|} [v(K \cup \{i\}) - v(K)] \end{aligned}$$

$$\begin{aligned}
&= \sum_{L \subseteq S} \sum_{K \subseteq L} (-1)^{|L|-|K|} \Delta v_i(K) \\
&= \sum_{K \subseteq S} \sum_{P \subseteq S \setminus K} (-1)^{|P|} \Delta v_i(K) \quad \% \text{ Let } P = L \setminus K \\
&= \sum_{K \subseteq S} \left( \sum_{p=0}^{|S|-|K|} \binom{|S|-|K|}{p} (-1)^p \right) \Delta v_i(K) \quad \% \text{ Let } p = |P| \\
&= \sum_{K \subseteq S} \left[ (1 + (-1)^{|S|-|K|}) \right] \Delta v_i(K) \quad \% \text{ Let } p = |P| \\
&= \sum_{K \subsetneq S} 0 \cdot \Delta v_i(K) + \sum_{K=S} \left( \sum_{p=0}^{|S|-|K|} \binom{|S|-|K|}{p} (-1)^p \right) \Delta v_i(K) \\
&= \Delta v_i(S) = \text{left}
\end{aligned}$$

**Connection between multi-order interactions and multi-order Shapley values.** Similarly, Equation (6) in the main paper also shows that the  $m$ -order Shapley value can also be decomposed as the sum of interaction benefits. This section provides the proof for such decomposition.

$$\phi_i^{(m)} = \mathbb{E}_{\substack{S \subseteq N \setminus \{i\} \\ |S|=m}} \left[ \sum_{L \subseteq S} I(L \cup \{i\}) \right] \quad (13)$$

• *Proof:*

$$\begin{aligned}
\phi_i^{(m)} &= \mathbb{E}_{S \subseteq N, |S|=m} \Delta v_i(S) \\
&= \mathbb{E}_{S \subseteq N, |S|=m} \left[ \sum_{L \subseteq S} I(L \cup \{i\}) \right] \\
&= \mathbb{E}_{S \subseteq N, |S|=m} \left[ \sum_{L \subseteq S} I(L \cup \{i\}) \right]
\end{aligned}$$

## E MORE EXPERIMENTAL RESULTS AND DETAILS

### E.1 DISCUSSION ABOUT EFFECTS OF INCORRECT BASELINE VALUES.

This section proves that the incorrect setting of baseline values makes a model/function consisting of high-order interaction patterns be mistakenly explained as a mixture of low-order and high-order interaction patterns. This is mentioned in Section 3.3 of the paper. To show this phenomenon, we compare interaction patterns computed using ground-truth baseline values and incorrect baseline values in Table 6, and the results verify our conclusion. We find that when models/functions contain complex collaborations between multiple variables (*i.e.* high-order interaction patterns), incorrect baseline values usually generate fewer high-order interaction patterns and more low-order interaction patterns than ground-truth baseline values. In other words, the model/function is explained as massive low-order interaction patterns. In comparison, ground-truth baseline values lead to sparse and high-order salient patterns.

We can understand the effects of incorrect baseline values as follows. When we use ground-truth baseline values, the absence of any variable will inactivate the interaction pattern. However, when we use incorrect baseline values, replacing an input variable  $i$  with its baseline value cannot completely remove the original information, or may bring in new information to activate new abnormal patterns. In other words, the variable  $i$  still provides meaningful information to the output. Therefore, incorrect baseline values cannot represent the absence state of variables, thus damaging the trustworthiness of the computed Shapley values.

Table 6: Comparison between ground-truth baseline values and incorrect baseline values. The last column shows ratios of multi-variate interaction patterns of different orders  $r_m = \frac{\sum_{S \subseteq N, |S|=m} |I(S)|}{\sum_{S \subseteq N, S \neq \emptyset} |I(S)|}$ . We consider interactions of input samples that activate interaction patterns. We find that when models/functions contain a single complex collaborations between multiple variables (*i.e.* high-order interaction patterns), incorrect baseline values usually generate a mixture of many low-order interaction patterns. In comparison, ground-truth baseline values lead to sparse and high-order interaction patterns.

Functions ( $\forall i \in N, i \in \{0, 1\}$ )	Baseline values $\mathbf{b}$	Ratios $\mathbf{r}$
$f(x) = x_1 x_2 x_3 x_4 x_5$ $x = [1, 1, 1, 1, 1]$	ground truth: $\mathbf{b}^* = [0, 0, 0, 0, 0]$ incorrect: $\mathbf{b}^{(1)} = [0.5, 0.5, 0.5, 0.5, 0.5]$ incorrect: $\mathbf{b}^{(2)} = [0.1, 0.2, 0.6, 0.0, 0.1]$ incorrect: $\mathbf{b}^{(3)} = [0.7, 0.1, 0.3, 0.5, 0.1]$	
$f(x) = \text{sigmoid}(5x_1 x_2 x_3 + 5x_4 - 7.5)$ $x = [1, 1, 1, 1]$	ground truth: $\mathbf{b}^* = [0, 0, 0, 0]$ incorrect: $\mathbf{b}^{(1)} = [0.5, 0.5, 0.5, 0.5]$ incorrect: $\mathbf{b}^{(2)} = [0.6, 0.4, 0.7, 0.3]$ incorrect: $\mathbf{b}^{(3)} = [0.3, 0.6, 0.5, 0.8]$	
$f(x) = x_1(x_2 + x_3 - x_4)^3$ $x = [1, 1, 1, 0]$	ground truth: $\mathbf{b}^* = [0, 0, 0, 1]$ incorrect: $\mathbf{b}^{(1)} = [0.5, 0.5, 0.5, 0.5]$ incorrect: $\mathbf{b}^{(2)} = [0.2, 0.3, 0.6, 0.1]$ incorrect: $\mathbf{b}^{(3)} = [1.0, 0.3, 1.0, 0.1]$	

## E.2 THE VISUALIZATION METHOD IN FIGURE 2.

This section provides experimental details of the visualization of Figure 2 in the paper. In Figure 2, the pixel value in the heatmap is computed as  $p(j|i) = \mathbb{E}_{S \in \Omega} [\mathbb{1}_{j \in S}]$ , where  $i \in N$  and  $\Omega$  denotes the set of patterns  $S$  whose values  $|\Delta v_i(S)|$  are relatively large among all  $S \subseteq N \setminus \{i\}$ .  $\Delta v_i(S)$  is defined in Equation (6).

Due to the exponential computational cost of enumerating all interaction patterns to obtain the salient ones in  $\Omega$ , we used the following greedy strategy for approximation. We first randomly sample an input variable  $i$  and a context  $S \subseteq N \setminus \{i\}$ , and compute the corresponding value of  $|\Delta v_i(S)|$ . We sample multiple times and find the pair of  $(i, S)$  that yields the largest value of  $|\Delta v_i(S)|$ . Then, we add/remove some variables to/from the context  $S$  to obtain a larger  $|\Delta v_i(S)|$ . Alg. 1 shows the pseudo-code of this algorithm.

## E.3 OTHER POTENTIAL SETTINGS OF $v(S)$ .

In the computation of Shapley values, people usually use different settings of  $v(S)$ , although the settings of  $v(S)$  do not affect the applicability of our method. Our method of formulating baseline values is applicable to various settings of  $v(S)$ . Lundberg & Lee (2017) directly set  $v(S) = p(y^{\text{truth}} | \text{mask}(x, S))$ . Covert et al. (2020b) used the cross-entropy loss as  $v(S)$ . In this paper, we use  $v(S) = \log \frac{p(y^{\text{truth}} | \text{mask}(x, S))}{1 - p(y^{\text{truth}} | \text{mask}(x, S))}$  in  $L_{\text{Shapley}}$ . Besides, we use  $|\Delta v_i(S)| = \|h(\text{mask}(x, S \cup \{i\}) - h(\text{mask}(x, S))\|_1$  in  $L_{\text{marginal}}$  on the MNIST dataset to boost the optimization efficiency, where  $h(\text{mask}(x, S))$  denotes the intermediate-layer feature. It is because  $h(\text{mask}(x, S \cup \{i\}))$  makes the optimization of  $L_{\text{marginal}}$  receive gradients from all dimensions of the feature.

## E.4 EXPERIMENTAL RESULTS ON THE UCI SOUTH GERMAN CREDIT DATASET.

This section provides experimental results on the UCI South German Credit dataset (Asuncion & Newman, 2007). Figure 7 shows the learned baseline values by our method, and Figure 8 compares Shapley values computed using different baseline values. Just like results on the UCI Census Income dataset, attributions (Shapley values) generated by our learned baseline values are similar to results of the varying baseline values in SHAP and SAGE. However, the zero/mean baseline values usually generated conflicting results with all other methods.



---

**Algorithm 1:** The approximate algorithm to interaction patterns whose  $\Delta v_i(S)$  are large

---

**Input:** The set of input variables  $N$ , the reward function (model)  $v(\cdot)$ , the sampling number of contexts  $t$ , the sampling number of noisy variables  $m$ , the convergence threshold  $\epsilon$ , the number  $k_{\max}$

**Output:** a specific input variable  $i$ , a set  $\Omega$  consisting of  $k_{\max}$  interaction patterns

```

1 for all  $i \in N$  do
2   Randomly sample a set of subsets  $\{P_1, P_2, \dots, P_t\} \subseteq 2^{N \setminus \{i\}}$ 
3    $S_i = \arg \max_P |\Delta v_i(P)|$ 
4 end
5  $i_{\max}, S_{\max} = \arg \max_{(i, S_i)} |\Delta v_i(S_i)|$ 
6 Initialize  $\Omega = \emptyset$ 
7 Randomly sample a set of subsets  $\{M_1, M_2, \dots, M_{k_{\max}}\} \subseteq 2^{N \setminus S_{\max} \setminus \{i_{\max}\}}$  with the size  $m$ , i.e.
    $\forall k \in \{1, 2, \dots, k_{\max}\}, |M_k| = m$ 
8 for  $k$  from 1 to  $k_{\max}$  do
9   Let  $S^{(k)} = S_{\max} \cup M_k, d^{(k)} = |\Delta v_{i_{\max}}(S^{(k)})|$ 
10  Initial  $S_{\max}^{(k)} = S^{(k)}, d_{\max}^{(k)} = d^{(k)}$ 
11  while True do
12    Let  $d_{\text{tmp}}^{(k)} = d_{\max}^{(k)}$ 
13    for all  $j \in N \setminus \{i\}$  do
14      if  $j \in S_{\max}^{(k)}$  then
15        if  $|\Delta v_{i_{\max}}(S_{\max}^{(k)} \setminus \{j\})| > d_{\max}^{(k)}$  then
16           $S_{\max}^{(k)} \leftarrow S_{\max}^{(k)} \setminus \{j\}$ 
17           $d_{\max}^{(k)} \leftarrow |\Delta v_{i_{\max}}(S_{\max}^{(k)} \setminus \{j\})|$ 
18        end
19      else
20        if  $|\Delta v_{i_{\max}}(S_{\max}^{(k)} \cup \{j\})| > d_{\max}^{(k)}$  then
21           $S_{\max}^{(k)} \leftarrow S_{\max}^{(k)} \cup \{j\}$ 
22           $d_{\max}^{(k)} \leftarrow |\Delta v_{i_{\max}}(S_{\max}^{(k)} \cup \{j\})|$ 
23        end
24      end
25    end
26    if  $|d_{\max}^{(k)} - d_{\text{tmp}}^{(k)}| < \epsilon$  then
27      Break
28    end
29  end
30   $\Omega \leftarrow \Omega \cup \{S_{\max}^{(k)}\}$ 
31 end
32 return  $i_{\max}, \Omega$ 

```

---

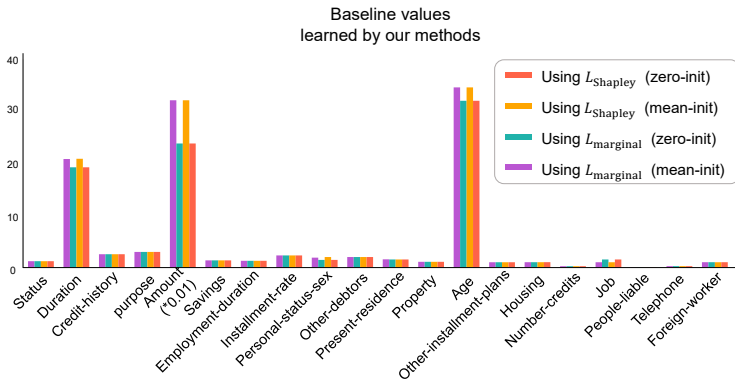


Figure 7: The learned baseline values on the UCI South German Credit dataset.

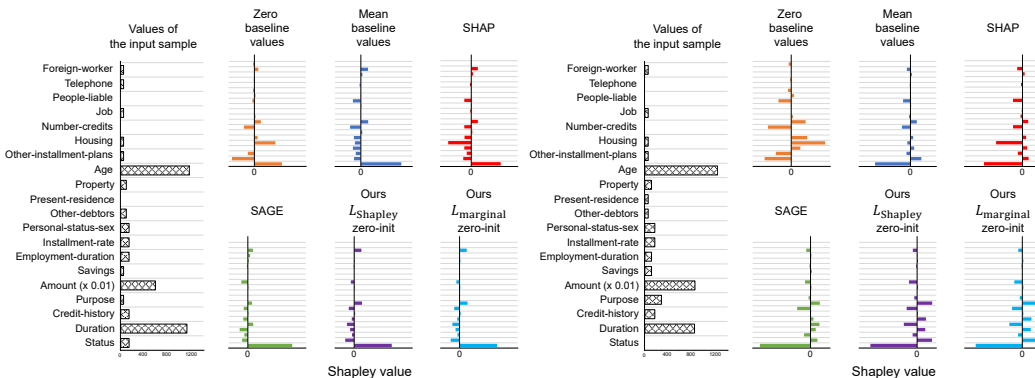


Figure 8: Shapley values computed with different baseline values on the UCI South German Credit dataset.

### E.5 STABILITY OF THE ESTIMATED SHAPLEY VALUES

In this section, we conducted an experiment to evaluate the stability of the estimated Shapley values via the sampling-based approximation method (Castro et al., 2009). Given a certain input sample, we fixed the sampling number and repeatedly computed the Shapley value multiple times. Then, we measured the instability of the estimated Shapley value  $\phi_i$  for a certain input variable  $i$ , which indicated that whether we could obtain similar Shapley values considering the randomness in each sampling process. The instability was computed as  $\frac{E_{u,v; u \neq v} |\phi_i^{(u)} - \phi_i^{(v)}|}{E_w |\phi_i^{(w)}|}$ , where  $\phi_i^{(u)}$  denote the estimated Shapley value in the  $u$ -th time. Then, we computed the average instability value over Shapley values of all variables in 20 input samples. We used MLPs learned from the UCI South German Credit dataset (Asuncion & Newman, 2007) and the UCI Census Income dataset. We computed the instability when we used different sampling numbers. As Figure 9 shows, on both datasets, the instability of the estimated Shapley values decreased along with the increase of the sampling number. We found that when the sampling number was larger than 1000, the instability of the estimated Shapley values was near or lower than 0.1, which meant the stable computed Shapley values. Therefore, we set the sampling number to 1000 in all other experiments in this paper.

### E.6 DISCUSSION ABOUT THE SETTING OF GROUND-TRUTH BASELINE VALUES.

This section discusses the ground truth of baseline values of synthetic functions in Section 4 of the paper. In order to verify the correctness of the learned baseline values, we conducted experiments on synthetic functions with ground-truth baseline values. We randomly generated 100 functions whose interaction patterns and ground truth of baseline values could be easily determined. As Table 7 shows,

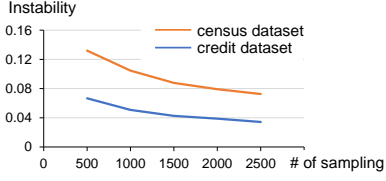


Figure 9: Instability of the Shapley values approximated with different sampling numbers. The instability of the approximated Shapley values decreased along with the increase of the sampling number.

Table 7: Examples of synthetic functions and their ground-truth baseline values.

Functions ( $\forall i \in N, x_i \in \{0, 1\}$ )	The ground truth of baseline values
$-0.185x_1(x_2 + x_3)^{2.432} - x_4x_5x_6x_7x_8x_9x_{10}x_{11}$	$b_i^* = 0$ for $i \in \{1, \dots, 11\}$
$-\text{sigmoid}(-4x_1 - 4x_2 - 4x_3 + 2) - 0.011x_4(x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11})^{2.341}$	$b_i^* = 1$ for $i \in \{1, 2, 3\}$ , $b_i^* = 0$ for $i \in \{4, \dots, 11\}$
$0.172x_1x_2x_3(x_4 + x_5)^{2.543} - 0.171x_6(x_7 + x_8)^{2.545} - 0.093(x_9 + x_{10} + x_{11})^{2.157}$	$b_i^* = 0$ for $i \in \{1, \dots, 11\}$
$-\text{sigmoid}(5x_1 + 5x_2x_3x_4 - 7.5) - \text{sigmoid}(-8x_5x_6 + 8x_7 + 4) + x_8x_9x_{10}x_{11}$	$b_i^* = 1$ for $i = 7$ , $b_i^* = 0$ for $i \in \{1, 2, 3, 4, 5, 6, 8, 9, 10, 11\}$
$-x_1x_2x_3 + 0.156(x_4 + x_5 + x_6)^{1.693} - x_7x_8x_9x_{10}$	$b_i^* = 0$ for $i \in \{1, \dots, 10\}$
$\text{sigmoid}(-3x_1x_2x_3 + 1.5) + 0.197x_4x_5(x_6 + x_7 + x_8)^{1.48} + x_9x_{10}x_{11}$	$b_i^* = 0$ for $i \in \{1, \dots, 11\}$
$-\text{sigmoid}(5x_1 - 5x_2x_3 + 2.5) - \text{sigmoid}(3x_4x_5x_6x_7 - 1.5) - 0.365x_8(x_9 + x_{10})^{1.453}$	$b_i^* = 1$ for $i = 1$ , $b_i^* = 0$ for $i \in \{2, \dots, 10\}$
$\text{sigmoid}(4x_1 - 4x_2 - 4x_3 + 6) - \text{sigmoid}(-6x_4x_5x_6x_7 + 3) - \text{sigmoid}(6x_8 - 6x_9x_{10} + 3)$	$b_i^* = 1$ for $i \in \{1, 8\}$ , $b_i^* = 0$ for $i \in \{2, 3, 4, 5, 6, 7, 9, 10\}$
$-x_1x_2x_3 + 0.205x_4(x_5 + x_6)^{2.289} - 0.115(x_7 + x_8 + x_9)^{1.969} + x_{10}x_{11}x_{12}$	$b_i^* = 0$ for $i \in \{1, \dots, 12\}$
$-\text{sigmoid}(-3x_1x_2x_3 - 3x_4x_5 + 4.5) - \text{sigmoid}(3x_6 + 3x_7 + 3x_8 - 1.5) + x_9x_{10}x_{11}$	$b_i^* = 1$ for $i \in \{6, 7, 8\}$ , $b_i^* = 0$ for $i \in \{1, \dots, 5, 9, 10, 11\}$
$-\text{sigmoid}(8x_1x_2 - 8x_3 - 8x_4 - 4) - 0.041(x_5 + x_6 + x_7 + x_8)^{2.298} - \text{sigmoid}(7x_9x_{10} + 7x_{11} - 10.5)$	$b_i^* = 1$ for $i \in \{3, 4\}$ , $b_i^* = 0$ for $i \in \{1, 2, 5, 6, \dots, 11\}$
$-\text{sigmoid}(4x_1 - 4x_2 + 4x_3 - 6) - x_4x_5x_6x_7 - x_8x_9x_{10}x_{11}$	$b_i^* = 1$ for $i = 2$ , $b_i^* = 0$ for $i \in \{1, 3, 4, \dots, 11\}$
$\text{sigmoid}(3x_1x_2 + 3x_3 - 3x_4 - 3x_5 - 3x_6 - 4.5) + \text{sigmoid}(6x_7x_8 + 6x_9x_{10}x_{11} - 9)$	$b_i^* = 1$ for $i \in \{4, 5, 6\}$ , $b_i^* = 0$ for $i \in \{1, 2, 3, 7, 8, 9, 10, 11\}$
$-\text{sigmoid}(-7x_1x_2 - 7x_3x_4 + 10.5) + \text{sigmoid}(-5x_5 + 5x_6 - 5x_7x_8 - 5x_9 + 12.5) + x_{10}x_{11}x_{12}$	$b_i^* = 1$ for $i = 6$ , $b_i^* = 0$ for $i \in \{1, 2, 3, 4, 5, 7, 8, \dots, 12\}$
$\text{sigmoid}(-6x_1 + 6x_2 + 6x_3 + 3) + 0.229x_4x_5x_6(x_7 + x_8)^{2.124} + 0.070x_9(x_{10} + x_{11} + x_{12})^{2.418}$	$b_i^* = 1$ for $i \in \{2, 3\}$ , $b_i^* = 0$ for $i \in \{1, 4, 5, \dots, 12\}$
$x_1x_2x_3x_4 - \text{sigmoid}(-6x_5x_6 - 6x_7 + 9) + x_8x_9x_{10}x_{11}$	$b_i^* = 0$ for $i \in \{1, \dots, 11\}$
$\text{sigmoid}(-3x_1 + 3x_2 - 3x_3x_4 - 3x_5x_6 + 7.5) - 0.174x_7(x_8 + x_9 + x_{10})^{1.594}$	$b_i^* = 1$ for $i = 2$ , $b_i^* = 0$ for $i \in \{1, 3, 4, \dots, 10\}$
$-0.34x_1(x_2 + x_3)^{1.557} - \text{sigmoid}(5x_4x_5x_6 - 5x_7 - 5x_8 - 5x_9 - 5x_{10} - 5x_{11} - 2.5)$	$b_i^* = 1$ for $i \in \{7, \dots, 11\}$ , $b_i^* = 0$ for $i \in \{1, \dots, 6\}$
$-\text{sigmoid}(-8x_1x_2 + 8x_3 + 4) - x_4x_5x_6x_7 + 0.457x_8(x_9 + x_{10})^{1.13}$	$b_i^* = 1$ for $i = 3$ , $b_i^* = 0$ for $i \in \{1, 2, 4, 5, 6, 7, 8, 9, 10\}$
$\text{sigmoid}(-6x_1 + 6x_2 - 6x_3 - 3) + \text{sigmoid}(-6x_4x_5 - 6x_6 + 6x_7 + 9) + \text{sigmoid}(4x_8x_9 - 4x_{10} - 2)$	$b_i^* = 1$ for $i \in \{1, 3, 7, 10\}$ , $b_i^* = 0$ for $i \in \{2, 4, 5, 6, 8, 9\}$

the generated functions were composed of addition, subtraction, multiplication, exponentiation, and *sigmoid* operations.

The ground truth of baseline values in these functions was determined based on interaction patterns between input variables. In order to represent absence states of variables, baseline values should activate as few salient patterns as possible, where activation states of interaction patterns were considered as the most infrequent state. Thus, we first identified the activation states of interaction patterns of variables, and the ground-truth of baseline values were set as values that inactivated interaction patterns under different masks. We took the following examples to discuss the setting of ground-truth baseline values (in the following examples,  $\forall i \in N, x_i \in \{0, 1\}$  and  $b_i^* \in \{0, 1\}$ ).

- $f(x) = x_1x_2x_3 + \text{sigmoid}(x_4 + x_5 - 0.5) + \dots$ . Let us just focus on the term of  $x_1x_2x_3$  in  $f(x)$ . The activation state of this interaction pattern is  $x_1x_2x_3 = 1$  when  $\forall i \in \{1, 2, 3\}, x_i = 1$ . In order to inactivate the interaction pattern, we set  $\forall i \in \{1, 2, 3\}, b_i^* = 0$ .

- $f(x) = -x_1x_2x_3 + (x_4 + x_5)^3 + \dots$ . Let us just focus on the term of  $-x_1x_2x_3$  in  $f(x)$ . The activation state of this interaction pattern is  $-x_1x_2x_3 = -1$  when  $\forall i \in \{1, 2, 3\}, x_i = 1$ . In order to inactivate the interaction pattern, we set  $\forall i \in \{1, 2, 3\}, b_i^* = 0$ .

- $f(x) = (x_1 + x_2 - x_3)^3 + \dots$ . Let us just focus on the term of  $(x_1 + x_2 - x_3)^3$  in  $f(x)$ . The activation state of this interaction pattern is  $(x_1 + x_2 - x_3)^3 = 8$  when  $x_1 = x_2 = 1, x_3 = 0$ . In order to inactivate the interaction pattern under different masks, we set  $b_1^* = b_2^* = 0, b_3^* = 1$ .

- $f(x) = \text{sigmoid}(3x_1x_2 - 3x_3 - 1.5) + \dots$ . Let us just focus on the term of  $\text{sigmoid}(3x_1x_2 - 3x_3 - 1.5)$  in  $f(x)$ . In this case,  $x_1, x_2, x_3$  form a salient interaction pattern because  $\text{sigmoid}(3x_1x_2 - 3x_3 - 1.5) > 0.5$  only if  $x_1 = x_2 = 1$  and  $x_3 = 0$ . Thus, in order to inactivate interaction patterns, ground-truth baseline values are set to  $b_1^* = b_2^* = 0, b_3^* = 1$ .

**Ground-truth baseline values of functions in (Tsang et al., 2018).** This section provides more details about ground-truth baseline values of functions proposed in (Tsang et al., 2018). We evaluated the correctness of the learned baseline values using functions proposed in (Tsang et al., 2018). Among all the 92 input variables in these functions, the ground truth of 61 variables could be determined and are reported in Table 8. Note that some variables cannot be 0 or 1 (e.g.  $x_8$  cannot be zero in the first function), and we set  $\forall i \in N, x_i \in \{0.001, 0.999\}$  for variables in these functions instead. Similarly, we set the ground truth of baseline values  $\forall i \in N, b_i^* \in \{0.001, 0.999\}$ . Some variables did not collaborate/interact with other variables (e.g.  $x_4$  in the first function), thereby having no interaction patterns. We did not assign ground-truth baseline values for these individual variables, and these variables are not used for evaluation. Some variables formed more than one interaction pattern with other variables, and had different ground-truth baseline values *w.r.t.* different patterns. In this case,

Table 8: Functions in (Tsang et al., 2018) and their ground-truth baseline values.

Functions ( $\forall i \in N, x_i \in \{0.001, 0.999\}$ )	The ground truth of baseline values
$\pi^{x_1 x_2} \sqrt{2x_3} - \sin^{-1}(x_4) + \log(x_3 + x_5) - \frac{x_6}{x_{10}} \sqrt{\frac{x_7}{x_8}} - x_2 x_7$	$b_i^* = 0.999$ for $i \in \{5, 8, 10\}$ , $b_i^* = 0.001$ for $i \in \{1, 2, 7, 9\}$
$\pi^{x_1 x_2} \sqrt{2 x_3 } - \sin^{-1}(0.5x_4) + \log( x_3 + x_5  + 1) + \frac{x_6}{1+ x_{10} } \sqrt{\frac{x_7}{1+ x_8 }} - x_2 x_7$	$b_i^* = 0.999$ for $i = 5$ , $b_i^* = 0.001$ for $i \in \{1, 2, 7, 9\}$
$\exp x_1 - x_2  +  x_2 x_3  - x_3^{2 x_4 } + \log(x_4^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2) + x_9 + \frac{1}{1+x_{10}}$	$b_i^* = 0.999$ for $i \in \{3, 5, 7, 8\}$
$\exp x_1 - x_2  +  x_2 x_3  - x_3^{2 x_4 } + (x_1 x_4)^2 + \log(x_4^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2) + x_9 + \frac{1}{1+x_{10}}$	$b_i^* = 0.999$ for $i \in \{3, 5, 7, 8\}$
$\frac{1}{1+x_1^2+x_2^2+x_3^2} + \sqrt{\exp(x_4 + x_5) +  x_6 + x_7  + x_8 x_9 x_{10}}$	$b_i^* = 0.999$ for $i \in \{1, 2, 3\}$ , $b_i^* = 0.001$ for $i \in \{4, 5, 8, 9, 10\}$
$\exp( x_1 x_2  + 1) - \exp( x_3 + x_4  + 1) + \cos(x_5 + x_6 - x_8) + \sqrt{x_8^2 + x_9^2 + x_{10}^2}$	$b_i^* = 0.999$ for $i \in \{8, 9, 10\}$ , $b_i^* = 0.001$ for $i \in \{1, 2, 3, 4, 5, 6\}$
$(\arctan(x_1) + \arctan(x_2))^2 + \max(x_3 x_4 + x_6, 0) - \frac{1}{1+(x_4 x_5 x_6 x_7 x_8)^2} + \left(\frac{ x_7 }{1+ x_9 }\right)^5 + \sum_{i=1}^{10} x_i$	$b_i^* = 0.999$ for $i = 9$ , $b_i^* = 0.001$ for $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$
$x_1 x_2 + 2^{x_3+x_4+x_5} + 2^{x_3+x_4+x_5+x_7} + \sin(x_7 \sin(x_8 + x_9)) + \arccos(0.9x_{10})$	$b_i^* = 0.001$ for $i \in \{1, 2, 3, 4, 5, 6\}$
$\tanh(x_1 x_2 + x_3 x_4) \sqrt{ x_5 } + \exp(x_5 + x_6) + \log((x_6 x_7 x_8)^2 + 1) + x_9 x_{10} + \frac{1}{1+ x_{10} }$	$b_i^* = 0.001$ for $i \in \{6, 7, 8, 9, 10\}$
$\sinh(x_1 + x_2) + \arccos(\tanh(x_3 + x_5 + x_7)) + \cos(x_4 + x_5) + \sec(x_7 x_9)$	$b_i^* = 0.999$ for $i = 3$ , $b_i^* = 0.001$ for $i \in \{1, 2, 4\}$

the collaboration between input variables was complex and hard to analyze, so we did not consider such input variables with conflicting patterns for evaluation, either.

## E.7 DISCUSSION ABOUT THE SETTING OF GROUND-TRUTH SHAPLEY VALUES.

This section discusses the ground truth of Shapley values in the extended Addition-Multiplication dataset (Zhang et al., 2021), which is used in Section 4 of the paper. In order to verify the correctness of the Shapley values obtained by the optimal baseline values in this paper, we conducted experiments on the extended Addition-Multiplication dataset (Zhang et al., 2021) with ground-truth Shapley values.

The Addition-Multiplication dataset in (Zhang et al., 2021) contained functions that only consisted of addition and multiplication operations. For example,  $f(x) = x_1 x_2 + x_3 x_4$  where each input variable  $x_i \in \{0, 1\}$  was a binary variable. Given  $x = [1, 1, 1, 1]$ , the function contained two salient interaction patterns, *i.e.*  $\{x_1, x_2\}$  and  $\{x_3, x_4\}$ , and their benefits to the output were  $I(\{x_1, x_2\}) = I(\{x_3, x_4\}) = 1$ , respectively. According to (Harsanyi, 1963), the Shapley value is a uniform distribution of attributions. Therefore, the benefit of an interaction pattern was supposed to be uniformly assigned to variables in the pattern. Thus, the ground-truth Shapley values of variables were  $\hat{\phi}_1 = \hat{\phi}_2 = 1/2$ , and  $\hat{\phi}_3 = \hat{\phi}_4 = 1/2$ . However, if the input  $x = [1, 0, 1, 1]$ , then the pattern  $\{x_1, x_2\}$  was deactivated and  $I(\{x_1, x_2\}) = 0$ . In this case,  $\hat{\phi}_1 = \hat{\phi}_2 = 0$  while  $\hat{\phi}_3 = \hat{\phi}_4 = 1/2$ .

According to the analysis in Appendix E.6, ground-truth baseline values in the Addition-Multiplication dataset were all zero. Then our method is equivalent to the zero baseline values. Therefore, in order to avoid all ground-truth baseline values being zero, we added the subtraction operation. We also added a coefficient before each term in the function to boost the diversity of functions. For example,  $f(x) = 3.2x_1 x_2 + 1.5x_3(x_4 - 1)$ . This function also contained two interaction patterns, but the ground-truth baseline values of variables were different from the aforementioned function. Here,  $b_0^* = b_0^* = b_3^* = 0$  and  $b_4^* = 1$ . Given the input  $x = [1, 1, 1, 0]$ , all patterns were activated and  $f(x) = I(\{x_1, x_2\}) + I(\{x_3, x_4\}) = 3.2 + (-1.5) = 1.7$ . Ground-truth Shapley values of input variables were  $\hat{\phi}_1 = \hat{\phi}_2 = 3.2/2$  and  $\hat{\phi}_3 = \hat{\phi}_4 = -1.5/2$ . However, for the input  $x = [1, 1, 1, 1]$ , the pattern  $\{x_3, x_4\}$  was deactivated, thereby  $\hat{\phi}_3 = \hat{\phi}_4 = 0$ . Note that the above function can also be considered to contain three patterns ( $f(x) = 3.2x_1 x_2 + 1.5x_3 x_4 - 1.5x_3$ ). According to Occam’s Razor, we follow the principle of the most simplified interaction to recognize interaction patterns in the function, *i.e.* using the least number of interaction patterns. Thus, we consider the above function  $f(x) = 3.2x_1 x_2 + 1.5x_3(x_4 - 1)$  containing two salient interaction patterns.

Based on the extended Addition-Multiplication dataset, we randomly generated an input sample for each function in the dataset. Each variable  $x_i$  in input samples were independently sampled following the Bernoulli distribution, *i.e.*  $p(x_i = 1) = 0.7$ . Therefore, for the mean baseline, baseline values of different input variables were all 0.7. For the baseline value based on the marginal distribution, which was used in SHAP (Lundberg & Lee, 2017),  $p(x_i') \sim \text{Bernoulli}(0.7)$ . Then, we compared the accuracy of the computed Shapley values of input variables based on zero baseline values, mean baseline values, baseline values in SHAP, and the optimal baseline values defined in this paper, respectively. The result in Table 4 shows that the optimal baseline values correctly generated the ground-truth attributions/Shapley values of input variables.

## E.8 DISCUSSION ABOUT THE BASELINE VALUES LEARNED ON ADVERSARIAL EXAMPLES.

In Section 4 of the main paper, we show that the learned baseline values can recover original samples from adversarial examples. This section provides more discussions about this experiment.

Let  $x$  denote the normal sample, and let  $x^{\text{adv}} = x + \delta$  denote the adversarial example generated by adversarial attacks (Madry et al., 2018). A previous study (Ren et al., 2021) defined the bivariate interaction with different contextual complexities, and found that adversarial attacks mainly created out-of-distribution bivariate interactions with large contexts. In the scenario of this study, we can consider such sensitive interactions with large contexts related to high-order multi-variate interaction patterns. It is because the bivariate interaction between  $(i, j)$  with large contexts (*i.e.* under many contextual variables  $S$ ) actually considers the collaboration between  $i, j$  and contextual variables in  $S$ . Thus, it also partially reflects  $I(S \cup \{i, j\})$ , which is defined in this paper.

Therefore, from the perspective of multi-variate interaction patterns, the adversarial utility can be considered as introducing out-of-distribution high-order interaction patterns in the model. To this end, setting input variables to baseline values is supposed to remove related interaction patterns to represent absence states. Particularly, if we set all input variables to their baseline values, many interaction patterns will be eliminated. Thus, if we initialize baseline values as the adversarial example, and optimize baseline values using our method, the learned baseline values are supposed to remove OOD high-order interaction patterns in the adversarial example, and recover the original sample.