Accelerating Shapley Explanation via Contributive Cooperator Selection

Guanchu Wang [*1] **Yu-Neng Chuang** [*1] **Mengnan Du** [2] **Fan Yang** [1]
Quan Zhou [3] **Pushkar Tripathi** [3] **Xuanting Cai** [3] **Xia Hu** [1]

Abstract

Even though Shapley value provides an effective explanation for a DNN model prediction, the computation relies on the enumeration of all possible input feature coalitions, which leads to the exponentially growing complexity. To address this problem, we propose a novel method SHEAR to significantly accelerate the Shapley explanation for DNN models, where only a few coalitions of input features are involved in the computation. The selection of the feature coalitions follows our proposed Shapley chain rule to minimize the absolute error from the ground-truth Shapley values, such that the computation can be both efficient and accurate. To demonstrate the effectiveness, we comprehensively evaluate SHEAR across multiple metrics including the absolute error from the ground-truth Shapley value, the faithfulness of the explanations, and running speed. The experimental results indicate SHEAR consistently outperforms state-of-the-art baseline methods across different evaluation metrics, which demonstrates its potentials in real-world applications where the computational resource is limited. The source code is available at https://github.com/guanchuwang/SHEAR.

1. Introduction

Despite the remarkable achievement of deep neural networks (DNNs) in a variety of fields, the black-box nature of DNNs still limits its deployment in domains where model explanations are required for acquiring trustful results, such as healthcare (Esteva et al., 2019), finance (Caruana et al., 2020) and recommender systems (Yang et al., 2018). Explaining the behavior of DNNs is a significant problem due to both the practical requirements of the stakeholders as well as the regulations in different domains, e.g., GDPR (Goodman et al., 2017; Floridi, 2019). To overcome the black-box nature of DNNs, existing work has developed various techniques for model interpretation such as gradient-based methods (Sundararajan et al., 2017), casual interpretation (Luo et al., 2020), counterfactual explanation (Yang et al., 2021) and Shapley value explanation (Lundberg & Lee, 2017). Among these, the Shapley value (Shapley, 2016) has emerged as a popular explanation approach due to its strong theoretical properties (Lipovetsky et al., 2001).

The Shapley value provides a natural and effective explanation for DNNs from the perspective of cooperative game theory (Kuhn & Tucker, 1953; Winter, 2002; Roth, 1988). The explanation can be adaptive to individual features, an instance, or the representation of global feature contribution (Covert et al., 2020). However, the Shapley explanation is known to be an NP-hard problem with extremely high computational complexity, which prevents its application to real-world scenarios. The brute-force algorithm to calculate the exact Shapley values requires the enumeration of all possible input feature coalitions, where the complexity grows exponentially with the feature number (Van D. B. et al., 2021). Hence, it is crucial to reduce the computational complexity of Shapley explanation. Existing work can be categorized into two groups to overcome this challenge. The first group studies specific approximation of Shapley value for DNN models (Chen et al., 2018; Ancona et al., 2019; Jia et al., 2019; Wang et al., 2021). Even though these kinds of work contribute to efficient explanations for the prediction of DNN models, they suffer from the inevitable gap with the ground-truth Shapley explanation due to the approximation (Liu et al., 2021), and lack of flexibility to deal with different types of models. Another group proposes the regression of Shapley values based on model evaluation on either the sampling of feature coalitions or permutations (Kokhlikyan et al., 2020; Covert et al., 2020). Although these kinds of methods provide the effective Shapley explanation for DNN models, they require large numbers of model evaluations, which is inefficient and often performs an undesirable trade-off between the computational complexity and interpretation performance (Covert

[*]Equal contribution [1]Department of Computer Science, Rice University [2]Department of Computer Science and Engineering, Texas A&M University [3]Meta Platforms, Inc.. Correspondence to: Guanchu Wang <guanchu.wang@rice.edu>, Xia Hu <xia.hu@rice.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

& Lee, 2021; Ribeiro et al., 2016; Liu et al., 2021).

In this work, we propose a novel method to reduce the complexity of Shapley value estimation for the acceleration of DNN explanation. Instead of using all possible input feature coalitions, we focus on selecting a few features to generate the coalitions for the estimation such that the complexity can be significantly reduced. Specifically, we first propose the Shapley chain rule to indicate that the absolute estimation error is related to the selection of input features, and those can optimize the estimation are termed as *contributive cooperators*. Then, we propose SHapley Explanation AcceleRation (SHEAR) following the Shapley chain rule to conduct contributive cooperator selection in the estimation of Shapley value. In this way, the enumeration of all possible feature coalitions can be avoided and the explanation achieves significant acceleration. To demonstrate SHEAR enables both accurate and efficient Shapley explanation, we conduct experiments on three benchmark datasets to compare it with five state-of-the-art baseline methods using five evaluation metrics. The evaluation involves two metrics to compare the explanation with ground-truth Shapley values: the absolute estimation error and accuracy of feature importance ranking (Wojtas & Chen, 2020); two metrics to evaluate the explanation via model perturbation: the faithfulness (Liu et al., 2021) and monotonicity (Luss et al., 2019); and algorithmic throughput (Teich & Teich, 2018) to evaluate the running speed. The contributions of this work are summarized as follows:

- We propose the Shapley chain rule to theoretically guide the complexity reduction of Shapley value estimation.
- Following the Shapley chain rule, we further propose SHEAR for Shapley explanation acceleration.
- Experimental results over three datasets indicate that our SHEAR works more efficiently than state-of-the-art methods without degradation of interpretation performance.

2. Preliminaries

In this section, we introduce the notations used throughout this work and provide an overview of Shapley values.

2.1. Notations

We consider an arbitrary DNN model f and input feature $\boldsymbol{x} \in \mathcal{X}$, where $x_1, \cdots, x_M$ denotes the value of input feature $1, \cdots, M$, respectively, and each feature has either continuous or categorical value. To formalize the contribution of each feature to the prediction, the inference of f is regarded as a cooperative game on the feature set $\mathcal{U} = \{1, \cdots, M\}$, where we let $f_v : 2^M \to \mathbb{R}$ denote the value function. Specifically, for a feature coalition (i.e., subset) $\boldsymbol{S} \subseteq \mathcal{U}$, $f_v(\boldsymbol{S})$ returns the prediction based on the features in coalition $\boldsymbol{S}$ that are marginalized over the features not in coalition $\boldsymbol{S}$, which is given as follows

$$f_v(\boldsymbol{S}) = \mathbb{E}\big[f(x_1, \cdots, x_M) \mid \boldsymbol{x}_{\mathcal{U}\setminus\boldsymbol{S}} \sim p(\boldsymbol{x}_{\mathcal{U}\setminus\boldsymbol{S}})\big], \quad (1)$$

where $\boldsymbol{x}_{\mathcal{U}\setminus\boldsymbol{S}}$ denotes $[x_j \mid j \in \mathcal{U} \setminus \boldsymbol{S}]$, and $p(\boldsymbol{x}_{\mathcal{U}\setminus\boldsymbol{S}})$ denotes the joint distribution of x_j for $j \in \mathcal{U} \setminus \boldsymbol{S}$.

However, the marginalized value function is difficult to estimate following Equation (1), since it depends on the enumeration of data instances over the whole dataset. A widely used solution (Lundberg & Lee, 2017; Wang et al., 2021; Kokhlikyan et al., 2020; Covert & Lee, 2021) is to approximate Equation (1) as follows

$$f_v(\boldsymbol{S}) \approx f\big(\boldsymbol{x}_{\boldsymbol{S}}, \overline{\boldsymbol{x}}_{\mathcal{U}\setminus\boldsymbol{S}}\big), \quad (2)$$

where $\boldsymbol{x}_{\boldsymbol{S}}$ denotes $[x_j \mid j \in \boldsymbol{S}]$; and $\overline{\boldsymbol{x}}_{\mathcal{U}\setminus\boldsymbol{S}} = \mathbb{E}[\boldsymbol{x}_{\mathcal{U}\setminus\boldsymbol{S}} \mid \boldsymbol{x}_{\mathcal{U}\setminus\boldsymbol{S}} \sim p(\boldsymbol{x}_{\mathcal{U}\setminus\boldsymbol{S}})]$ denotes the reference values[1] of feature $j \in \mathcal{U} \setminus \boldsymbol{S}$. We follow this approximation in this work.

2.2. Shapley Value

Shapley value regards the input features to the DNN model as the *cooperators*, and estimates the feature contribution from the perspective of cooperative game theory (Kuhn & Tucker, 1953). Specifically, it assigns an importance value $\phi_i(f_v, \mathcal{U})$ to indicate the contribution of feature $i \in \mathcal{U}$ to the DNN prediction, and formalizes the feature contribution following the brute-force algorithm in Equation (3). According to Equation (3), Shapley value adopts the preceding difference $f_v(\{i\} \cup \boldsymbol{S}) - f_v(\boldsymbol{S})$ to indicate the contribution of feature i considering the features in coalition $\boldsymbol{S}$, and enumerates the feature coalitions throughout the cooperators, which are the M input features. The average preceding difference considering all possible feature coalitions indicates the contribution of feature i,

$$\phi_i(f_v, \mathcal{U}) = \frac{1}{M} \sum_{\boldsymbol{S} \subseteq \mathcal{U}\setminus\{i\}} \binom{M-1}{|\boldsymbol{S}|}^{-1} \big[f_v(\{i\} \cup \boldsymbol{S}) - f_v(\boldsymbol{S})\big]. \quad (3)$$

The brute-force algorithm takes all of the M features as the cooperators, and relies on 2^M times of model evaluation to estimate the contribution of feature i, which has the computational complexity exponentially growing with the feature number $T[\phi_i(f, \mathcal{U})] = O(2^M)$. In this work, we propose a low-complexity estimation of the feature contribution $\{\hat{\phi}_i\}_{1 \le i \le M}$, where the contribution of each feature $\hat{\phi}_i$ is estimated based on N times of model evaluation, and we have $N \ll 2^M$. The estimation aims to minimize the following absolute error taking the ground-truth Shapley value (GT-Shapley value) as the reference,

$$\min \sum_{i=1}^{M} |\phi_i(f_v, \mathcal{U}) - \hat{\phi}_i|, \quad (4)$$

where $\hat{\phi}_i$ denotes the estimated contribution of feature i.

[1] Other statistic value can also be adopted for the reference value.

3. Shapley Chain Rule

In this section, we propose *Shapley Chain Rule* in Theorem 1 to provide theoretical instructions for the Shapley explanation acceleration. Note that the brute-force algorithm considers all of the M input features as the cooperators, which leads the complexity of $\phi_i(f_v,\mathcal{U})$ to grow exponentially with the feature number. Shapley chain rule provides the estimation of feature contribution based on only a few features as the cooperators to significantly reduce the computational complexity. We give the proof of Theorems 1 and 2 in Appendix B and C, respectively.

Theorem 1 (Shapley Chain Rule)**.** *For any differentiable value function $f_v : 2^M \to \mathbb{R}$, the contribution of feature i to $f_v(\mathcal{U})$ satisfies*

$$\phi_i(f_v,\mathcal{U}) = \phi_i(f_v,\mathcal{U}\setminus\{j\}) + \Delta_{i,j} + o_{i,j},$$

where $j \in \mathcal{U}\setminus\{i\}$ denotes another feature; $\phi_i(f_v,\mathcal{U}\setminus\{j\})$ denotes the contribution of feature i to $f_v(\mathcal{U}\setminus\{j\})$; $o_{i,j}=o(x_i-\bar{x}_i)+o(x_j-\bar{x}_j)$; and the error term $\Delta_{i,j}$ is given by

$$\Delta_{i,j} = (x_i-\bar{x}_i)(x_j-\bar{x}_j) \sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \frac{\nabla^2_{i,j}f_v(\boldsymbol{S}\cup\{i,j\})+\nabla^2_{j,i}f_v(\boldsymbol{S}\cup\{i,j\})}{2(M-|\boldsymbol{S}|-1)\binom{M}{|\boldsymbol{S}|+1}},$$

where x_i denotes the value of feature i; $\bar{x}_i$ denotes the reference value of feature i; and $\nabla^2_{i,j}f_v(\boldsymbol{S}) = \frac{\partial^2 f(\boldsymbol{x}_{\boldsymbol{S}},\bar{\boldsymbol{x}}_{\mathcal{U}\setminus\boldsymbol{S}})}{\partial x_i \partial x_j}$ denotes the cross-gradient towards x_i and x_j.

Remark 1 (Complexity Reduction). The computational complexity of $\phi_i(f_v,\mathcal{U}\setminus\{j\})$ equals to the half of $\phi_i(f_v,\mathcal{U})$.

According to Remark 1, the computational complexity can be significantly reduced if we remove feature j from the cooperators such that the contribution of feature i can be estimated by $\phi_i(f_v,\mathcal{U}\setminus\{j\})$. However, it causes a significant estimation error according to Theorem 1. To reduce the complexity but without loss of accuracy, we propose Theorem 2 to bound the error term.

Theorem 2 (Upper Bound of Error Term)**.** *For any features $i \neq j \in \mathcal{U}$, the upper bound of the absolute gap between $\phi_i(f_v,\mathcal{U})$ and $\phi_i(f_v,\mathcal{U}\setminus\{j\})$ is given by*

$$\left|\phi_i(f_v,\mathcal{U})-\phi_i(f_v,\mathcal{U}\setminus\{j\})\right| \le \epsilon_{i,j}|x_i-\bar{x}_i||x_j-\bar{x}_j|, \quad (5)$$

where $\epsilon_{i,j}$ relies on the gradient towards x_i and x_j by

$$\epsilon_{i,j} = \max_{\mathcal{V}\subseteq\mathcal{U}\setminus\{i,j\}} \frac{1}{4}\left|\nabla^2_{i,j}f_v(\mathcal{U}\setminus\mathcal{V})+\nabla^2_{j,i}f_v(\mathcal{U}\setminus\mathcal{V})\right|. \quad (6)$$

Remark 2. For any feature subset $\mathcal{S}\subseteq\mathcal{U}\setminus\{i\}$, the absolute gap between $\phi_i(f_v,\mathcal{U})$ and $\phi_i(f_v,\mathcal{S}\cup\{i\})$ is bounded by

$$\left|\phi_i(f_v,\mathcal{U})-\phi_i(f_v,\mathcal{S}\cup\{i\})\right| \le \sum_{j\in\mathcal{U}\setminus\mathcal{S}\setminus\{i\}} \epsilon_{i,j}|x_i-\bar{x}_i||x_j-\bar{x}_j|. \quad (7)$$

Note that Equation (7) provides the upper bound for the absolute estimation error in Equation (4). We take $\hat{\phi}_i = \phi_i(f,\mathcal{S}_i\cup\{i\})$, for $1\le i\le M$. In this way, the problem of reducing the estimation complexity can be formulated into selecting the optimal contributive cooperators for each feature to minimize the worst-case absolute estimation error. According to the upper bound in Equation (7), we have

$$\mathcal{S}_i = \arg\min_{\mathcal{S}\subseteq\mathcal{U}\setminus\{i\}} |\phi_i(f_v,\mathcal{U}) - \phi_i(f_v,\mathcal{S}\cup\{i\})|, \quad (8)$$

$$\sim \arg\min_{\mathcal{S}\subseteq\mathcal{U}\setminus\{i\}} \sum_{j\in\mathcal{U}\setminus\mathcal{S}} \epsilon_{i,j}|x_i-\bar{x}_i||x_j-\bar{x}_j|, \quad (9)$$

where we have $|\mathcal{S}_i| = \log_2\frac{N}{2}$ to constrain the number of contributive cooperators such that the estimation complexity satisfies $T[\phi_i(f,\mathcal{S}_i\cup\{i\})] = O(N)$, where N denotes the time of model evaluation which depends on the available limited computational resource.

For the convenience of optimization, we transform the $\arg\min$ problem in Equation (9) to $\arg\max$ as the objective function of the contributive cooperator selection as follows,

$$\mathcal{S}_i = \arg\max_{\substack{\mathcal{S}\subseteq\mathcal{U}\setminus\{i\}\\ |\mathcal{S}|=\log_2(N/2)}} \sum_{j\in\mathcal{S}} \epsilon_{i,j}|x_i-\bar{x}_i||x_j-\bar{x}_j|. \quad (10)$$

4. Shapley Explanation Acceleration

In this section, we propose the SHapley Explanation AcceleRation (SHEAR) to provide efficient Shapley explanations for DNN models. First, we propose the definition of feature *cross-contribution* for SHEAR to approximately solve Equation (10). Second, we adopt antithetical sampling for SHEAR to promote the estimation. Afterwards, we give the details of SHEAR. Finally, the empirical complexity analysis is provided for SHEAR.

4.1. Feature Cross-contribution

The selection of contributive cooperators is challenging following Equation (10) due to the high computational complexity of $\epsilon_{i,j}$ following Equation (6). To address this problem, we propose to approximate $\epsilon_{i,j}$ into $\hat{\epsilon}_{i,j} \approx \frac{1}{4}|\nabla^2_{i,j}f_v(\mathcal{U}) + \nabla^2_{j,i}f_v(\mathcal{U})|$, and propose feature *cross-contribution* in Definition 1 for reaching the approximately optimal solution of Equation (10) in SHEAR.

Definition 1 (Cross-contribution). For features $i\neq j\in\mathcal{U}$, the cross-contribution of features i and j is defined as

$$\eta_{i,j} = |x_i-\bar{x}_i|\left|\nabla^2_{i,j}f_v(\mathcal{U}) + \nabla^2_{j,i}f_v(\mathcal{U})\right||x_j-\bar{x}_j|, \quad (11)$$

where $\nabla^2_{i,j}f_v(\mathcal{U}) = \frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}$ denotes the cross-gradient[2] towards features i and j; and we have the cross-gradient satisfying $\nabla^2_{i,j}f_v(\mathcal{U}) = \nabla^2_{j,i}f_v(\mathcal{U})$ for DNNs.

Remark 3. Definition 1 considers a special case $\mathcal{V}=\varnothing$ for Equation (6) to simplify the computation.

Intuitively, $\eta_{i,j}$ indicates the strength of cooperation be-

[2] torch.autograd provides APIs to estimate the backward gradient of DNNs.

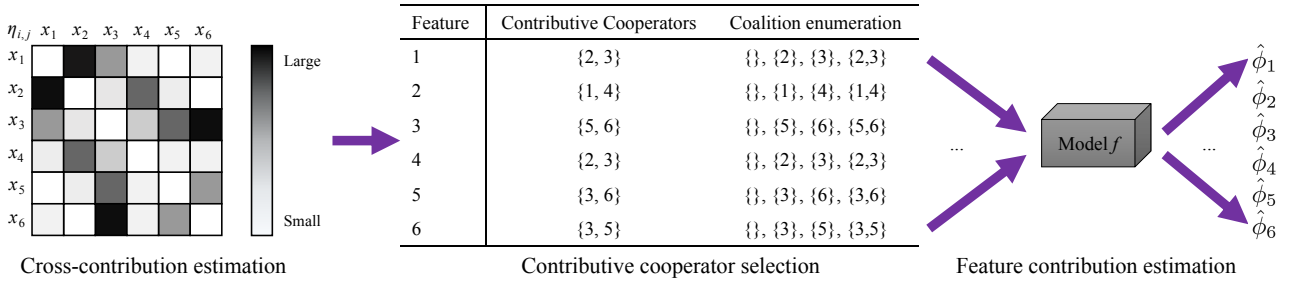


Figure 1: Algorithmic configuration of SHEAR.

tween features i and j; and the feature subset that can maximize the cross-contribution with feature i provides an approximately optimal solution for Equation (10). In such a manner, SHEAR selects the contributive cooperators $\mathcal{S}_i$ for each feature $i \in \mathcal{U}$ following

$$\mathcal{S}_i = \underset{\substack{\mathcal{S}\subseteq\mathcal{U}\setminus\{i\} \\ |\mathcal{S}|=\log_2(N/2)}}{\arg\max} \sum_{j\in\mathcal{S}} \eta_{i,j}, \tag{12}$$

where we have the constraint $|\mathcal{S}| = \log_2 \frac{N}{2}$ for the $\arg\max$ problem such that $T(\hat{\phi}_i) = O(N)$; and the optimal solution of Equation (12) can be achieved via ranking and greedy search under $O(M \log M)$ complexity. After this, SHEAR estimates the contribution of feature i following

$$\hat{\phi}_i = \frac{1}{|\mathcal{S}_i|+1} \sum_{\boldsymbol{S}\subseteq\mathcal{S}_i} \binom{|\mathcal{S}_i|}{|\boldsymbol{S}|}^{-1} \left[f_v(\{i\}\cup\boldsymbol{S}) - f_v(\boldsymbol{S}) \right]. \tag{13}$$

4.2. Antithetical Sampling

Note that Equation (13) totally ignores the influence of non-contributive features $j \in \mathcal{U}\setminus\mathcal{S}_i\setminus\{i\}$ to the value function, which leads to sub-optimal solutions. To address this problem but without extra calculation, SHEAR employs the antithetical sampling (AS) (Lomeli et al., 2019) to fix the preceding difference in Equation (13) to promote the estimation. Specifically, following the enumeration of $\boldsymbol{S} \subseteq \mathcal{S}_i$ in Equation (13) where $|\mathcal{S}_i| = \log_2 \frac{N}{2}$, let $\boldsymbol{S}_1, \cdots, \boldsymbol{S}_{N/2} \subseteq \mathcal{S}_i$ denote the entirely possible feature coalitions (i.e. subsets) on $\mathcal{S}_i$ where $\boldsymbol{S}_1 \neq \cdots \neq \boldsymbol{S}_{N/2}$. In each case of the enumeration where $\boldsymbol{S} = \boldsymbol{S}_n$, the AS revises the preceding difference $f_v(\{i\}\cup\boldsymbol{S}_n) - f_v(\boldsymbol{S}_n)$ into $f_v(\{i\}\cup\boldsymbol{S}_n\cup\boldsymbol{V}_n) - f_v(\boldsymbol{S}_n\cup\boldsymbol{V}_n)$ such that the contribution of feature i is estimated as follows

$$\hat{\phi}_i = \frac{1}{|\mathcal{S}_i|+1} \sum_{\boldsymbol{S}_n\subseteq\mathcal{S}_i} \binom{|\mathcal{S}_i|}{|\boldsymbol{S}|}^{-1} \left[f_v(\{i\}\cup\boldsymbol{S}_n\cup\boldsymbol{V}_n) - f_v(\boldsymbol{S}_n\cup\boldsymbol{V}_n) \right], \tag{14}$$

where $\boldsymbol{V}_n$ satisfies $\boldsymbol{V}_n \cup \boldsymbol{V}_{n'} \subseteq \mathcal{U}\setminus\mathcal{S}_i\setminus\{i\}$ for $1 \leq n, n' \leq \frac{N}{2}$ and $n + n' = \frac{N}{2} + 1$.

An example is given in Table 1 for $M = 6$, $i = 1$ and $N = 8$. Assume the optimization following Euqation (12) achieves $\mathcal{S}_1 = \{2, 3\}$. The enumeration of $\boldsymbol{S} \subseteq \mathcal{S}_1$ following Equation (14) involves $\boldsymbol{S}_1 = \varnothing$, $\boldsymbol{S}_2 = \{2\}$, $\boldsymbol{S}_3 = \{3\}$ and $\boldsymbol{S}_4 = \{2, 3\}$. SHEAR randomly samples the subsets of non-contributive features $\boldsymbol{V}_1 = \{5\}$ and $\boldsymbol{V}_2 = \{5, 6\}$ from $\mathcal{U}\setminus\mathcal{S}_1\setminus\{1\} = \{4, 5, 6\}$; then adopts the AS to have $\boldsymbol{V}_3 = \mathcal{U}\setminus\mathcal{S}_1\setminus\{1\}\setminus\boldsymbol{V}_2 = \{4\}$ and $\boldsymbol{V}_4 = \mathcal{U}\setminus\mathcal{S}_1\setminus\{1\}\setminus\boldsymbol{V}_1 = \{4, 6\}$; finally estimates $\hat{\phi}_1$ following Equation (14).

Table 1: An example of antithetical sampling.

$\mathcal{U}$	i	$\mathcal{S}_i$	n	$\boldsymbol{S}_n$	$\boldsymbol{V}_n$
{ 1,2,3,4,5,6 }	1	{ 2,3 }	1	$\varnothing$	{ 5 }
			2	{2}	{ 5,6 }
			3	{3}	{4}
			4	{2,3}	{ 4,6 }

4.3. Algorithm of SHEAR

The configuration and pseudo code of SHEAR are given in Figure 1 and Algorithm 1, respectively. To be concrete, SHEAR receives a DNN model f and feature value $x_1, \cdots, x_M$, and outputs the contributions $\hat{\phi}_1, \cdots, \hat{\phi}_M$ of feature $1, \cdots, M$, respectively. For each feature i, SHEAR first calculates its cross-contribution $\eta_{i,j}$ with other features $j \neq i \in \mathcal{U}$ following Equation (11) (Line 2); then greedily selects the contributive cooperators $\mathcal{S}_i$ following Equation (12) (Line 3) to maximize the cross-contribution $\sum_{j\in\mathcal{S}_i} \eta_{i,j}$; finally estimates the contribution of feature i (Line 4) throughout the coalitions of contributive cooperators $\boldsymbol{S} \subseteq \mathcal{S}_i$ following Equation (14).

SHEAR enables the estimation of feature contribution to avoid the enumeration of all possible feature coalitions. In this way, the estimation complexity can be significantly reduced from the brute-force complexity $T[\phi_i(f_v, \mathcal{U})] = O(2^M)$ to $T(\hat{\phi}_i) = O(N)$, where $N \ll 2^M$. Moreover, the estimation process for the M features can execute independently without dependency on each other, which can be deployed on distributed systems for speeding up.

Algorithm 1 SHapley Explanation AcceleRation (SHEAR)

Input: DNN model f, input values $\boldsymbol{x} = [x_1, \cdots, x_M]$.
Output: Estimation value of feature contribution $\hat{\phi}_1, \ldots, \hat{\phi}_M$.

1: **for** $i = 1, 2, \cdots, M$ **do**
2: Estimate $\eta_{i,j}$ following Equation (11) for $j \in \mathcal{U}\setminus\{i\}$.
3: Select contributive cooperators $\mathcal{S}_i$ via Equation (12).
4: Estimate feature contribution $\hat{\phi}_i$ by Equation (14).
5: **end for**

4.4. Time Consumption Analysis

In this section we analyze the time consumption of SHEAR. Generally, the time consumption of backward and forward process of DNNs is much more than that of other operators. Hence, we only focus on the time-cost of model forward or backward processes in SHEAR, and ignore the arithmetic and comparison operators in our analysis. According to Algorithm 1, SHEAR has one backward process to calculate the gradient $\nabla^2_{i,j} f_v(\mathcal{U}) + \nabla^2_{j,i} f_v(\mathcal{U})$ for the estimation of cross-contribution, and $2^{|\mathcal{S}_i|} \times 2 = N$ forward processes for the estimation of feature contribution following Equation (14). Hence, the estimation for a single feature $\hat{\phi}_i$ has the time consumption given by

$$T_{\text{SHEAR}} \approx t_{\text{backward}} + N t_{\text{forward}}, \tag{15}$$

where t_{backward} and t_{forward} denote the time consumption of model backward and forward process, respectively.

Considering the interpretation of a model f that has M input features, the overall time cost increases to M times if the M features are processed consecutively; or we can reduce the time cost by simultaneously processing the M input features based on the parallel structure. We consider the first case in our experiments because this work focuses on algorithmic acceleration instead of the engineering trick. To indicate the running speed of SHEAR, we analyze the algorithmic throughput in Section 5.4.

5. Experiment

In this section, we conduct experiments to evaluate SHEAR by answering the following research questions. In comparison to state-of-the-art baseline methods, **RQ1**: Does SHEAR provide more accurate explanation by comparing with GT-Shapley value? **RQ2**: Does SHEAR provide more faithful explanations? **RQ3**: Does SHEAR run faster than the baseline methods? **RQ4**: Does the contributive cooperator selection following Equation (12) contribute to SHEAR?

5.1. Experiment Setup

We provide the details about benchmark datasets, baseline methods and the experiment pipeline in this section.

Dataset: The experiments involve Census Income, German Credit and Cretio datasets from the areas of social media, finance and recommender systems, respectively. More details about the datasets are provided in Appendix F.

Baseline Methods: SHEAR is compared with five state-of-the-art baseline methods of Shapley value estimation, including Kernel-SHAP (KS) (Lundberg & Lee, 2017), Kernel-SHAP with Welford algorithm (KS-WF) (Covert & Lee, 2021), Kernel-SHAP with Pair Sampling (KS-Pair) (Covert & Lee, 2021), Permutation Sampling (PS) (Mitchell et al., 2021) and Antithetical Permutation Sampling (APS) (Lomeli et al., 2019). More details about the baseline methods are provided in Appendix G.

Implementation Details: The experiments on each dataset follow the pipeline of *model training*: training the DNN model; *interpretation benchmark*: adopting the brute-force algorithm to calculate the exact Shapley value as the ground truth explanation for the evaluation; and *interpretation evaluation*: evaluating the interpretation methods. We give the details about each step in Appendix H.

5.2. Evaluation with GT-Shapley Value (RQ1)

In this section, we evaluate the interpretation methods via taking the GT-Shapley value as the ground truth explanation, and taking the GT-Shapley value ranking as the ground truth feature importance ranking. Specifically, we consider two metrics to evaluate the interpretation performance: the absolute estimation error (AE) and accuracy of feature importance ranking (ACC). Given the GT-Shapley value $\phi_1, \cdots, \phi_M$ from the brute-force algorithm, the metrics AE and ACC are formulated as follows:

Absolute estimation error: For the estimation value of feature contribution $\hat{\phi}_1, \cdots, \hat{\phi}_M$, the absolute estimation error is given by

$$\text{AE} = \sum_{i=1}^{M} \mid \phi_i - \hat{\phi}_i \mid .$$

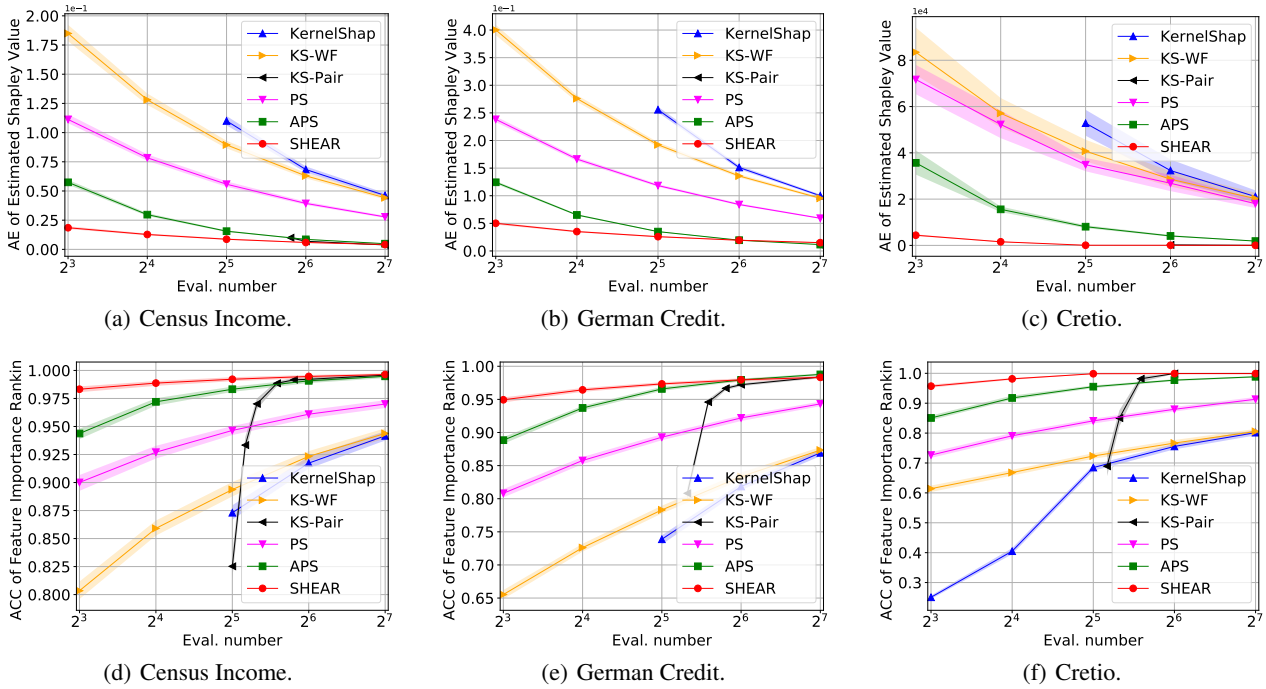
Accuracy of feature importance ranking: Let $r_1, \cdots, r_M$ and $\hat{r}_1, \cdots, \hat{r}_M$ denote the descending ranking of $\phi_1, \cdots, \phi_M$ and $\hat{\phi}_1, \cdots, \hat{\phi}_M$, respectively. The accuracy of feature importance ranking (Wojtas & Chen, 2020) can be calculated as follows:

$$\text{ACC} = \frac{\sum_{m=1}^{M} \frac{\mathbf{1}_{\hat{r}_m = r_m}}{m}}{\sum_{m=1}^{M} \frac{1}{m}},$$

where the factor $\frac{1}{m}$ enables the important features contribute more to the accuracy; and the factor $(\sum_{m=1}^{M} \frac{1}{m})^{-1}$ normalizes the accuracy such that $0 \leq \text{ACC} \leq 1$.

We give the absolute estimation error versus the times of model evaluation in Figures 2 (a)-(c), and the accuracy of feature importance ranking in Figures 2 (d)-(f); we also plot the error bar to show the standard deviation of each method across multiple rounds. According to the experimental results, we have the following observations:

- As the number of model evaluation grows, we observe less absolute estimation error and more accurate feature importance ranking of each method on the three datasets. The reason is that more model evaluations enable each method to converge to the GT-Shapley value.
- With unified model evaluation time, SHEAR achieves the least absolute estimation error and the most accurate feature importance ranking. The experimental results indicate the efficient utilization of model evaluations en-



(a) Census Income. (b) German Credit. (c) Cretio.

(d) Census Income. (e) German Credit. (f) Cretio.

Figure 2: Absolute estimation error of the feature contribution on the (a) Census Income, (b) German Credit and (c) Cretio datasets; Accuracy of feature importance ranking on the (d) Census Income, (e) German Credit and (f) Cretio datasets.

ables SHEAR to adapt to real-world application, where the available times of model evaluation is far from enough to enumerate all possible input feature coalitions.

- Compared with Kernel-SHAP, KS-WF and PS, SHEAR gets a WIN-WIN situation, where SHEAR shows MORE accurate explanation using FEWER model evaluations.
- SHEAR shows the least standard deviation of the absolute error and accuracy compared with the baseline methods due to the fact that SHEAR adopts greedy search to select the contributive cooperators without random initialization. The stable interpretation performance of SHEAR indicates its robustness towards different models trained on different datasets.

5.3. Evaluation with Model Perturbation (RQ2)

The evaluation methods with model perturbation are motivated by the common sense that important features have more impact on the prediction than trivial features. Specifically, we follow the existing work (Liu et al., 2021) to consider two metrics to evaluate the interpretation performance with model perturbation: *Faithfulness* and *Monotonicity*.

Faithfulness: For the estimation value of feature contribution $\hat{\phi}_1, \cdots, \hat{\phi}_M$, Faithfulness computes the Pearson correlation coefficient (Benesty et al., 2009) between the feature contribution and preceding difference of the model,

$$\text{Faithful} = \text{Pearson}\Big(\big[f_v(\mathcal{U}) - f_v(\mathcal{U}\backslash\{i\})\big]_{1\le i\le M}, [\hat{\phi}_i]_{1\le i\le M}\Big),$$

where the Pearson correlation coefficient is given by: $\text{Pearson}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$; $\text{cov}(X,Y)$ denotes the covariance of X and Y; and σ_X and σ_Y denote the standard deviation of X and Y, respectively. Larger Faithfulness indicates better interpretation.

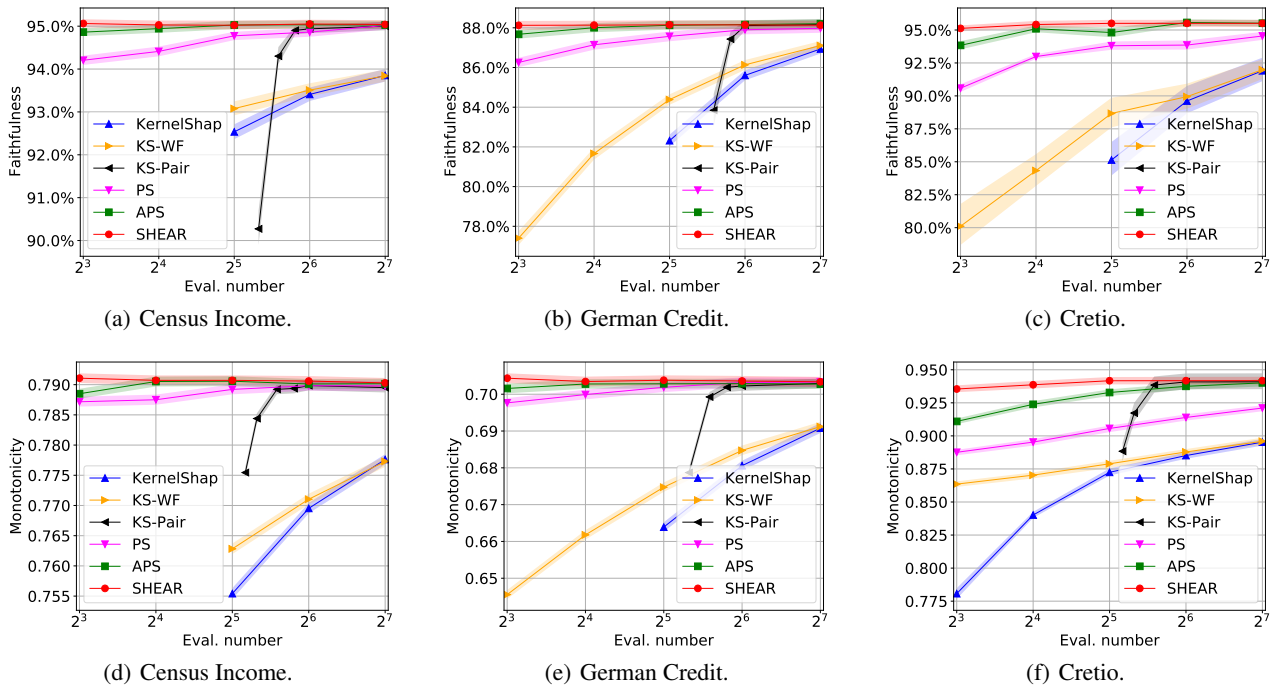
Monotonicity: Monotonicity evaluates the feature importance ranking without the ground-truth ranking. It computes the marginal improvement of each feature ordered by the estimated feature contribution, and calculates the fraction of indices i such that the marginal improvement of feature i is greater than feature $i+1$. Monotonicity is given by

$$\text{Monotonicity} = \frac{1}{M-1}\sum_{i=0}^{M-2} \mathbf{1}_{\delta_i \ge \delta_{i+1}},$$

where $\delta_i = f_v(\mathcal{W}_i \cup \{i\}) - f_v(\mathcal{W}_i)$ indicates the marginal improvement of the top-i important feature; and $\mathcal{W}_i = \{j \in \mathcal{U}\backslash\{i\} \mid \phi_j \ge \phi_i\}$ denotes the features more important than the top-i important feature. Larger Monotonicity implies better feature importance ranking.

The Faithfulness of SHEAR and baseline methods versus the times of model evaluation are shown in Figures 3 (a)-(c), and Monotonicity versus the times of model evaluation are give in Figures 3 (d)-(f). The error bar is given in the figures to show the standard deviation of each method. Overall, we have the following observations:

- All methods achieve larger Faithfulness and Monotonicity as the times of model evaluation grows due to the fact that more evaluations provide more information for



(a) Census Income. (b) German Credit. (c) Cretio.

(d) Census Income. (e) German Credit. (f) Cretio.

Figure 3: Faithfulness of the explanation on the (a) Census Income, (b) German Credit and (c) Cretio datasets; Monotonicity of the explanation on the (d) Census Income, (e) German Credit and (f) Cretio datasets.

interpreting the prediction.

- SHEAR achieves larger Faithfulness and Monotonicity than baseline methods under the same times of model evaluation, which indicates the effectiveness of SHEAR.
- SHEAR shows consistently better performance than baseline methods as the times of model decreases.
- According to Figures 2 and 3, even though the curves of absolute error, accuracy of feature importance ranking, Faithfulness and Monotonicity are different, the interpretation performance ranking of SHEAR and baseline methods indicated by the four metrics are consistent with each other. Hence, the effectiveness of SHEAR can be demonstrated by multiple evaluation metrics.

5.4. Throughput Evaluation (RQ3)

Algorithmic throughput is estimated by $\frac{N_{\text{test}}}{t_{\text{total}}}$, where N_{test} and t_{total} denote the testing instance number and the total time consumption of the interpreting process, respectively. N_{test} of the three datasets is given in Appendix F, and t_{total} is tested based on the physical computing infrastructure given in Appendix I. We plot the throughput versus the accuracy of feature importance ranking on the three datasets in Figures 4 (a)-(c), respectively, where we omit the curve of Kernel-SHAP due to its low accuracy observed from previous experiments. According to the experimental results, we have the following observations:

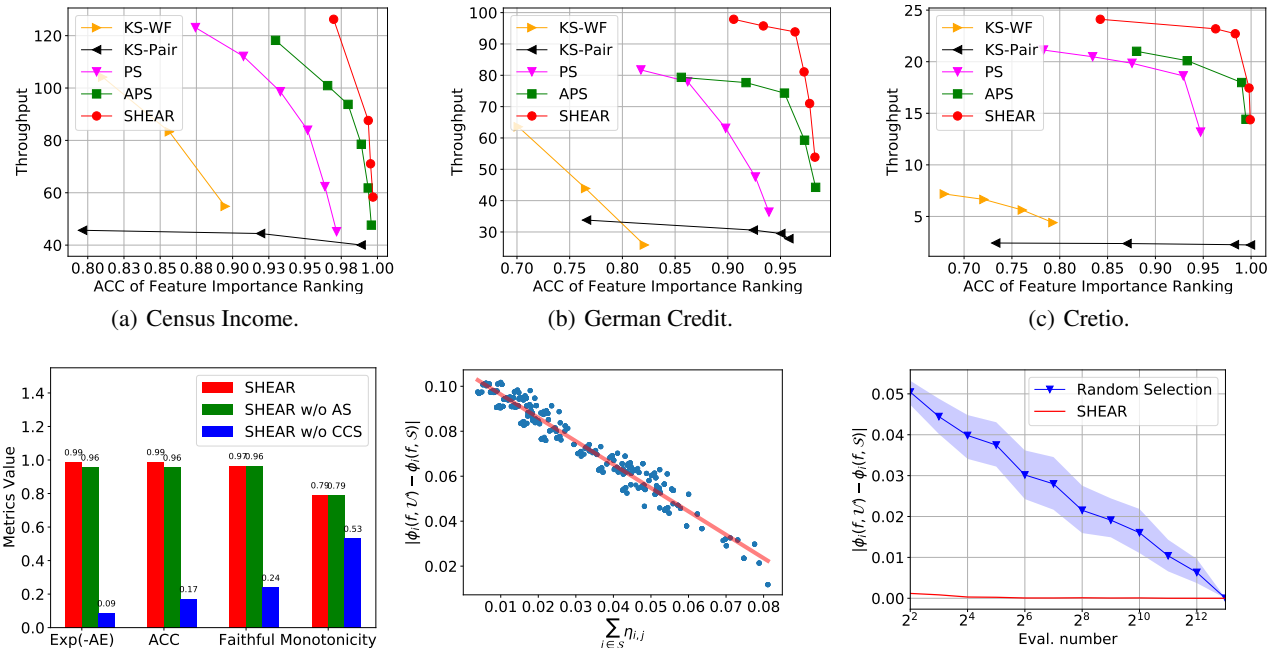
- For all methods, the throughput reduces as the interpretation performance grows due to the fact that more accurate explanation depends on more model evaluations which spend more time in the model forward process.
- SHEAR shows the most efficient interpretation, which has the highest accuracy when controlling the throughput and the most throughput when controlling the accuracy.
- Even though SHEAR has additional time cost on the gradient estimation according to Equation (15), it provides more accurate explanations than baseline methods, thus having better throughput and accuracy trade-off.

5.5. Ablation Study (RQ4)

The effectiveness of SHEAR mainly derives from the contributive cooperator selection following Equation (12). To prove this, we compare the interpretation performance of SHEAR, SHEAR without antithetical sampling (AS) and SHEAR without cooperator selection (CCS) in Figure 4 (d), where $N = 16$. We have the following observations:

- SHEAR outperforms SHEAR w/o AS, which demonstrates the effectiveness of antithetical sampling.
- SHEAR w/o CCS shows considerable interpretation degradation across the four evaluation metrics, which indicates the dominant contribution of CCS to SHEAR.

We conduct the follow-up experiments to trace the contributive cooperator selection. Specifically, for each instance in the testing dataset, we select a target feature $i \in \mathcal{U}$ to calculate the absolute error $|\phi_i(f, \mathcal{U}) - \phi_i(f, \mathcal{S} \cup \{i\})|$ via the brute-force algorithm, and calculate the cross-contribution $\sum_{j \in \mathcal{S}} \eta_{i,j}$ following Equation (11), where



(a) Census Income. (b) German Credit. (c) Cretio.

(d) Ablation study. (e) Estimation error vs. Cross-contribution. (f) Estimation error vs. Evaluation times.

Figure 4: Algorithmic throughput vs. ACC of feature importance ranking on the (a) Census Income, (b) German Credit and (c) Cretio datasets; (d) SHEAR vs. SHEAR w/o AS vs. SHEAR w/o CCS; (e) Absolute estimation error vs. Cross-contribution; (f) Absolute estimation error vs. Times of model evaluation.

$\mathcal{S}\subseteq\mathcal{U}\setminus\{i\}$ is randomly selected satisfying $|\mathcal{S}|=\log\frac{N}{2}$. For $N=16$, we illustrate the numerical relationship between $|\phi_i(f,\mathcal{U})-\phi_i(f,\mathcal{S}\cup\{i\})|$ and $\sum_{j\in\mathcal{S}}\eta_{i,j}$ in Figure 4 (e); and for $2^2\leq N\leq 2^M$, we plot the $|\phi_i(f,\mathcal{U})-\phi_i(f,\mathcal{S})|$ versus the evaluation times N in Figure 4 (f). According to the experimental results, we have the following observations:

- Approximately negative relationship between the absolute error $|\phi_i(f,\mathcal{U})-\phi_i(f,\mathcal{S})|$ and the feature cross-contribution $\sum_{j\in\mathcal{S}}\eta_{i,j}$ can be observed in Figure 4 (e).
- According to Figure 4 (f), we have $|\phi_i(f,\mathcal{U})-\phi_i(f,\mathcal{S})|$ drops from maximum value to 0 as N grows, where more features are involved to be the cooperators. Meanwhile, SHEAR achieves the lower bound of absolute error.

The above observations indicate the contributive cooperator selection of SHEAR following Equation (12) contributes to the minimization of the absolute estimation error, thus leading to accurate estimation of the feature contribution.

5.6. Illustration of Explanation

We illustrate the Shapley explanation generated by SHEAR for the DNN model prediction on the Census Income dataset to demonstrate the application of SHEAR. Specifically, we follow the settings in Appendix H to train the DNN model f. After the training, we select an instance $\tilde{\boldsymbol{x}}=[\tilde{x}_1,\cdots,\tilde{x}_M]$ from the testing dataset, and have $f_v(\mathcal{U})=f^1(\tilde{\boldsymbol{x}})-f^0(\tilde{\boldsymbol{x}})=1.73$, where $\text{softmax}[f(\boldsymbol{x})][i]$ denotes the predicted probability of the instance $\boldsymbol{x}$ belonging to class i. We have $i\in\{0,1\}$ for the income prediction task on the Census Income dataset, where $i=0$ or 1 indicates an instance has income *more than* or *less than* 50K/yr, respectively. The model outputs $\hat{y}=\text{sgn}[f^1(\tilde{\boldsymbol{x}})-f^0(\tilde{\boldsymbol{x}})]=1$ which indicates $\tilde{\boldsymbol{x}}$ has the income *more than* 50K/yr; and the base value of the model on the testing dataset satisfies $f_v(\varnothing)=f(\tilde{\boldsymbol{x}}_{\mathcal{U}})=-1.88$. We illustrate the Shapley explanation generated by SHEAR ($N=16$) in Figure 5 of Appendix A, where the GT-Shapley value is also shown for comparison. Overall, we have the following observations:

- The summation of the M feature contributions equals the distance between base value and model output, i.e. $\sum_{i=1}^{M}\hat{\phi}_i(f_v,\mathcal{U})=f_v(\mathcal{U})-f_v(\varnothing)=3.61$, where $M=13$ for the Census Income dataset.
- The base value $f_v(\varnothing)<0$ due to the fact that the classifier f is learned based on unbalanced training dataset with more negative samples than positive samples.
- The top-three key features for the income prediction are *education*, *martial-status* and *occupation*.
- The prediction of income has gender bias where *gender=male* has positive contribution to the prediction.

6. Conclusion

In this work, we propose the Shapley chain rule and SHEAR for the acceleration of Shapley explanation. The Shapley chain rule provides the theoretical instructions to minimize the absolute estimation error, and SHEAR follows the chain

rule to accelerate the Shapley explanation via contributive cooperator selection. The experimental results on three datasets using five evaluation metrics demonstrate that our proposed SHEAR works more efficiently than state-of-the-art methods without degradation of interpretation performance, and indicate the potential application of SHEAR to the real-world scenarios where the computational resource is limited.

References

Ancona, M., Oztireli, C., and Gross, M. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pp. 272–281. PMLR, 2019.

Benesty, J., Chen, J., Huang, Y., and Cohen, I. Pearson correlation coefficient. In *Noise reduction in speech processing*, pp. 1–4. Springer, 2009.

Caruana, R., Lundberg, S., Ribeiro, M. T., Nori, H., and Jenkins, S. Intelligible and explainable machine learning: Best practices and practical challenges. In *Proceedings of the 26th ACM SIGKDD*, pp. 3511–3512, 2020.

Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*, 2018.

Covert, I. and Lee, S.-I. Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*, pp. 3457–3465. PMLR, 2021.

Covert, I., Lundberg, S. M., and Lee, S.-I. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33, 2020.

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., and Dean, J. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

Floridi, L. Establishing the rules for building trustworthy ai. *Nature Machine Intelligence*, 1(6):261–262, 2019.

Goodman, B., Flaxman, S., and X, Y. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57, 2017.

Guo, H., Tang, R., Ye, Y., Li, Z., and He, X. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. J. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176. PMLR, 2019.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

Kuhn, H. W. and Tucker, A. W. *Contributions to the Theory of Games*, volume 2. Princeton University Press, 1953.

Lipovetsky, S., Conklin, M., and X, Y. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.

Liu, Y., Khandagale, S., White, C., and Neiswanger, W. Synthetic benchmarks for scientific research in explainable machine learning. 2021.

Lomeli, M., Rowland, M., Gretton, A., and G., Z. Antithetic and monte carlo kernel estimators for partial rankings. *Statistics and Computing*, 29(5):1127–1147, 2019.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

Luo, Y., Peng, J., and Ma, J. When causal inference meets deep learning. *Nature Machine Intelligence*, 2(8):426–427, 2020.

Luss, R., Chen, P.-Y., Dhurandhar, A., Sattigeri, P., Zhang, Y., Shanmugam, K., and Tu, C.-C. Generating contrastive explanations with monotonic attribute functions. *arXiv preprint arXiv:1905.12698*, 2019.

Mitchell, R., Cooper, J., Frank, E., and Holmes, G. Sampling permutations for shapley value estimation. *arXiv preprint arXiv:2104.12199*, 2021.

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

Roth, A. E. Introduction to the shapley value. *The Shapley value*, pp. 1–27, 1988.

Shapley, L. S. *17. A value for n-person games*. Princeton University Press, 2016.

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

Teich, D. A. and Teich, P. R. Plaster: A framework for deep learning performance. Technical report, Tech. rep. TIRIAS Research, 2018.

Van D. B., G., Lykov, A., S., M., and S., D. On the tractability of shap explanations. In *Proceedings of AAAI*, 2021.

Wang, R., Wang, X., and Inouye, D. I. Shapley explanation networks. *arXiv preprint arXiv:2104.02297*, 2021.

Welford, B. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

Winter, E. The shapley value. *Handbook of game theory with economic applications*, 3:2025–2054, 2002.

Wojtas, M. and Chen, K. Feature importance ranking for deep learning. *arXiv preprint arXiv:2010.08973*, 2020.

Yang, F., Liu, N., Wang, S., and Hu, X. Towards interpretation of recommender systems with sorted explanation paths. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 667–676. IEEE, 2018.

Yang, F., Alva, S. S., Chen, J., and Hu, X. Model-based counterfactual synthesizer for interpretation. *arXiv preprint arXiv:2106.08971*, 2021.

Appendix

A. Illustration of Explanation

We illustrate the Shapley explanation generated by SHEAR for the DNN model prediction on the Census Income dataset in this section. Specifically, the details about training the DNN model can be referred to Appendix H. After training the DNN model f, we select an instance $\tilde{\boldsymbol{x}} = [\tilde{x}_1, \cdots, \tilde{x}_M]$ from the testing dataset, and have the value function given by $f_v(\mathcal{U}) = f^1(\tilde{\boldsymbol{x}}) - f^0(\tilde{\boldsymbol{x}}) = 1.73$, where $f^i(\boldsymbol{x})$ denotes the prediction of the instance $\boldsymbol{x}$ belonging to class i. For the selected instance $\tilde{\boldsymbol{x}}$, the DNN model outputs $\hat{y} = \text{sgn}[f^1(\tilde{\boldsymbol{x}}) - f^0(\tilde{\boldsymbol{x}})] = 1$; and the base value of the model on the testing dataset satisfies $f_v(\varnothing) = f(\bar{\boldsymbol{x}}_{\mathcal{U}}) = -1.88$. We illustrate the Shapley explanation generated by SHEAR ($N = 16$) in Figure 5, and give the GT-Shapley value for comparison. We have several observations on the results which can be referred to Section 5.6.

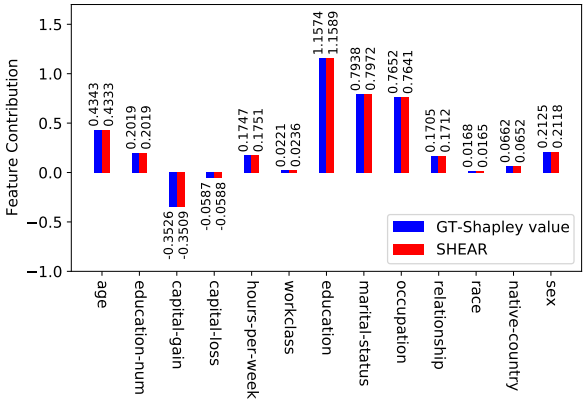


Figure 5: Explanation of model prediction for an instance from the Census Income dataset.

B. Proof of Theorem 1

In this section, we first propose Corollaries 1 and 2, then utilize the corollaries to prove Theorem 1.

Before proving the theorem, we propose Corollaries 1 and 2.

Corollary 1. *Given function $f(x_i, \cdots) : \mathcal{X} \to \mathbb{R}$ and $\bar{x}_i = \mathbb{E}_{x_i \sim p(x_i)}(x_i)$, $\forall (x_i, \cdots) \in \mathcal{X}$, $\exists$ function g such that*

$$f(x_i, \cdots) = f(\bar{x}_i, \cdots) + g(x_i, \cdots) + o(x_i - \bar{x}_i), \quad (16)$$

where '$\cdots$' is the abbreviation of non-active variables.

Proof. Without loss of generality, we adopt Taylor's theorem to expand $f(x_i, \cdots)$ at point $(\bar{x}_i, \cdots)$ as follows,

$$f(x_i, \cdots) = f(\bar{x}_i, \cdots) + \frac{\partial f}{\partial X_i}(x_i - \bar{x}_i) + \frac{1}{2}\frac{\partial^2 f}{\partial X_i^2}(x_i - \bar{x}_i)^2 + o(x_i - \bar{x}_i). \quad (17)$$

Let $g(x_i, \cdots) = \frac{\partial f}{\partial X_i}(x_i - \bar{x}_i) + \frac{1}{2}\frac{\partial^2 f}{\partial X_i^2}(x_i - \bar{x}_i)^2$. Take the $g(x_i, \cdots)$ into Equation (17), we achieve $f(x_i, \cdots) = f(\bar{x}_i, \cdots) + g(x_i, \cdots) + o(x_i - \bar{x}_i)$.

□

Corollary 2. *Given function $f(x_i, x_j, \cdots) : \mathcal{X} \to \mathbb{R}$ and $\bar{x}_i = \mathbb{E}_{x_i \sim p(x_i)}(x_i)$, $\forall (x_i, x_j, \cdots) \in \mathcal{X}$, $\exists$ functions g and $h : \mathcal{X} \to \mathbb{R}$ such that*

$$\begin{aligned} &f(x_i, x_j, \cdots) \\ &= f(\bar{x}_i, \bar{x}_j, \cdots) + g(x_i, \bar{x}_j, \cdots) + h(\bar{x}_i, x_j, \cdots) \\ &+ \lambda(x_i - \bar{x}_i)(x_j - \bar{x}_j) + o(x_i - \bar{x}_i) + o(x_j - \bar{x}_j), \end{aligned} \quad (18)$$

where $\lambda = \frac{1}{2}\left(\frac{\partial f}{\partial X_i \partial X_j} + \frac{\partial f}{\partial X_j \partial X_i}\right)\Big|_{X_i = x_i, X_j = x_j, \cdots}$.

Proof. Without loss of generality, we employ Taylor's theorem to expand $f(x_i, x_j, \cdots)$ at point $(\bar{x}_i, \bar{x}_j, \cdots)$ towards variables x_i and x_j as follows,

$$\begin{aligned} &f(x_i, x_j, \cdots) \\ &= f(\bar{x}_i, \bar{x}_j, \cdots) + \frac{\partial f}{\partial X_i}(x_i - \bar{x}_i) + \frac{1}{2}\frac{\partial^2 f}{\partial X_i^2}(x_i - \bar{x}_i)^2 \\ &+ \frac{\partial f}{\partial X_j}(x_j - \bar{x}_j) + \frac{1}{2}\frac{\partial^2 f}{\partial X_j^2}(x_j - \bar{x}_j)^2 \\ &+ \frac{1}{2}\left(\frac{\partial^2 f}{\partial X_i \partial X_j} + \frac{\partial^2 f}{\partial X_j \partial X_i}\right)(x_i - \bar{x}_i)(x_j - \bar{x}_j) \\ &+ o(x_i - \bar{x}_i) + o(x_j - \bar{x}_j). \end{aligned} \quad (19)$$

Equation (19) can be reformulated into $f(x_i, x_j, \cdots) = f(x_i, x_j, \cdots) + g(x_i, \bar{x}_j, \cdots) + h(\bar{x}_i, x_j, \cdots) + o(x_i - \bar{x}_i) + o(x_j - \bar{x}_j)$, where

$$\begin{aligned} g(x_i, \bar{x}_j, \cdots) &= \frac{\partial f}{\partial X_i}(x_i - \bar{x}_i) + \frac{1}{2}\frac{\partial^2 f}{\partial X_i^2}(x_i - \bar{x}_i)^2, \\ h(\bar{x}_i, x_j, \cdots) &= \frac{\partial f}{\partial X_j}(x_j - \bar{x}_j) + \frac{1}{2}\frac{\partial^2 f}{\partial X_j^2}(x_j - \bar{x}_j)^2. \end{aligned} \quad (20)$$

□

After Corollaries 1 and 2, we return to prove the theorem.

Theorem 1 (Shapley Chain Rule)**.** *For any differentiable value function $f_v : 2^M \to \mathbb{R}$, the contribution of feature i to $f_v(\mathcal{U})$ satisfies*

$$\phi_i(f_v, \mathcal{U}) = \phi_i(f_v, \mathcal{U} \setminus \{j\}) + \Delta_{i,j} + o_{i,j},$$

where $j \in \mathcal{U} \setminus \{i\}$ denotes another feature; $\phi_i(f_v, \mathcal{U} \setminus \{j\})$ denotes the contribution of feature i to $f_v(\mathcal{U} \setminus \{j\})$; $o_{i,j} = o(x_i - \bar{x}_i) + o(x_j - \bar{x}_j)$; and the error term $\Delta_{i,j}$ is given by

$$\Delta_{i,j} = (x_i - \bar{x}_i)(x_j - \bar{x}_j) \sum_{\boldsymbol{S} \subseteq \mathcal{U} \setminus \{i,j\}} \frac{\nabla^2_{i,j} f_v(\boldsymbol{S} \cup \{i,j\}) + \nabla^2_{j,i} f_v(\boldsymbol{S} \cup \{i,j\})}{2(M - |\boldsymbol{S}| - 1)\binom{M}{|\boldsymbol{S}|+1}}, \quad (21)$$

where x_i denotes the value of feature i; $\bar{x}_i$ denotes the reference value of feature i; and $\nabla^2_{i,j} f_v(\boldsymbol{S}) = \frac{\partial^2 f(\boldsymbol{x}_{\boldsymbol{S}}, \bar{\boldsymbol{x}}_{\mathcal{U}\setminus\boldsymbol{S}})}{\partial x_i \partial x_j}$ denotes the cross-gradient towards x_i and x_j.

Proof. For the simplicity of deduction but without loss of generality, we regard features i and j as the active variables and adopt the abbreviation $f_v(\{i\}\cup\boldsymbol{S}) = f(x_i, \bar{x}_j, \cdots)$ and $f_v(\boldsymbol{S}) = f(\bar{x}_i, \bar{x}_j, \cdots)$ for $\boldsymbol{S} \subseteq \mathcal{U}\setminus\{i,j\}$, where the non-active features are abbreviated into '$\cdots$'. Taking Corollary 1 into f_v, we have

$$\begin{aligned} &f_v(\{i\}\cup\boldsymbol{S}) - f_v(\boldsymbol{S}) \\ &= f_v(x_i, \bar{x}_j, \cdots) - f_v(\bar{x}_i, \bar{x}_j, \cdots) \\ &= g(x_i, \bar{x}_j, \cdots) + o(x_i - \bar{x}_i). \end{aligned} \tag{22}$$

Taking Equation (22) into Equation (3), we have the feature contribution $\phi_i(f, \mathcal{U}\setminus\{j\})$ given by

$$\begin{aligned} &\phi_i(f, \mathcal{U}\setminus\{j\}) \\ &= \frac{1}{M-1}\sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \binom{M-2}{|\boldsymbol{S}|}^{-1} \big[f_v(\{i\}\cup\boldsymbol{S}) - f_v(\boldsymbol{S})\big] \\ &= \frac{1}{M-1}\sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \binom{M-2}{|\boldsymbol{S}|}^{-1} g(x_i, \bar{x}_j, \cdots) + o(x_i - \bar{x}_i). \end{aligned} \tag{23}$$

Similarly, we have the abbreviation $f_v(\{i,j\}\cup\boldsymbol{S}) = f(x_i, x_j, \cdots)$ and $f_v(\{j\}\cup\boldsymbol{S}) = f(\bar{x}_i, x_j, \cdots)$ for $\boldsymbol{S} \subseteq \mathcal{U}\setminus\{i\}$. Taking Corollary 2 into f_v, we have

$$\begin{aligned} &f_v(\{i,j\}\cup\boldsymbol{S}) - f_v(\{j\}\cup\boldsymbol{S}) \\ &= f(x_i, x_j, \cdots) - f(\bar{x}_i, x_j, \cdots) \\ &= g(x_i, \bar{x}_j, \cdots) + \lambda_{i,j}(x_i - \bar{x}_i)(x_j - \bar{x}_j) \\ &+ o(x_i - \bar{x}_i) + o(x_j - \bar{x}_j), \end{aligned} \tag{24}$$

where $\lambda_{i,j} = \frac{1}{2}\Big(\frac{\partial f}{\partial X_i \partial X_j} + \frac{\partial f}{\partial X_j \partial X_i}\Big)\Big|_{X_i = x_i, X_j = x_j, \cdots}$.

According to Equation (3), the feature contribution $\phi_i(f, \mathcal{U})$ can be reformulated as follows,

$$\begin{aligned} \phi_i(f,\mathcal{U}) &= \frac{1}{M}\sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i\}} \binom{M-1}{|\boldsymbol{S}|}^{-1} \big[f_v(\{i\}\cup\boldsymbol{S}) - f_v(\boldsymbol{S})\big] \\ &= \frac{1}{M}\sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \binom{M-1}{|\boldsymbol{S}|}^{-1} \big[f_v(\{i\}\cup\boldsymbol{S}) - f_v(\boldsymbol{S})\big] \quad (25) \\ &+ \frac{1}{M}\sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \binom{M-1}{|\boldsymbol{S}|+1}^{-1} \big[f_v(\{i,j\}\cup\boldsymbol{S}) - f_v(\{j\}\cup\boldsymbol{S})\big]. \quad (26) \end{aligned}$$

Taking Equation (22) into Equation (25) and taking Equation (24) into Equation (26), we have

$$\begin{aligned} &\phi_i(f,\mathcal{U}) \\ &= \frac{1}{M}\sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \left[\binom{M-1}{|\boldsymbol{S}|}^{-1} + \binom{M-1}{|\boldsymbol{S}|+1}^{-1}\right] g(x_i, \bar{x}_j, \cdots) \quad (27) \\ &+ \frac{1}{M}\sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \binom{M-1}{|\boldsymbol{S}|+1}^{-1} \lambda_{i,j}(x_i - \bar{x}_i)(x_j - \bar{x}_j) \quad (28) \\ &+ o(x_i - \bar{x}_i) + o(x_j - \bar{x}_j). \quad (29) \end{aligned}$$

Theorem 1 can be proved via taking Equation (23) into Equation (27); transforming Equation (28) into Equation (21); and having $o_{i,j} = o(x_i - \bar{x}_i) + o(x_j - \bar{x}_j)$.

□

C. Proof of Theorem 2

We prove Theorem 2 in this section.

Theorem 2 (Upper Bound of Error Term)**.** *For any features $i \neq j \in \mathcal{U}$, the upper bound of the absolute gap between $\phi_i(f_v, \mathcal{U})$ and $\phi_i(f_v, \mathcal{U}\setminus\{j\})$ is given by*

$$\big|\phi_i(f_v,\mathcal{U}) - \phi_i(f_v,\mathcal{U}\setminus\{j\})\big| \leq \epsilon_{i,j}|x_i - \bar{x}_i||x_j - \bar{x}_j|, \tag{30}$$

where $\epsilon_{i,j}$ relies on the gradient towards x_i and x_j by

$$\epsilon_{i,j} = \max_{\mathcal{V}\subseteq\mathcal{U}\setminus\{i,j\}} \frac{1}{4}\big|\nabla^2_{i,j} f_v(\mathcal{U}\setminus\mathcal{V}) + \nabla^2_{j,i} f_v(\mathcal{U}\setminus\mathcal{V})\big|. \tag{31}$$

Proof. We ignore the infinitesimal $o_{i,j}$ in Theorem 1 to prove the upper bound. According to Theorem 1, we have

$$\big|\phi_i(f_v,\mathcal{U}) - \phi_i(f_v,\mathcal{U}\setminus\{j\})\big| \leq w_{i,j}|x_i - \bar{x}_i||x_j - \bar{x}_j|, \tag{32}$$

where $w_{i,j}$ is given by

$$w_{i,j} = \sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \frac{\nabla^2_{i,j} f_v(\boldsymbol{S}\cup\{i,j\}) + \nabla^2_{j,i} f_v(\boldsymbol{S}\cup\{i,j\})}{2(M - |\boldsymbol{S}| - 1)\binom{M}{|\boldsymbol{S}|+1}}.$$

Define $\epsilon_{i,j}$ following Equation (31), we have

$$w_{i,j} \leq \epsilon_{i,j}\sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \frac{2}{(M - |\boldsymbol{S}| - 1)\binom{M}{|\boldsymbol{S}|+1}} = \epsilon_{i,j}. \tag{33}$$

Eventually, Theorem 2 can be proved via taking Equation (33) to Equation (32), where

$$\begin{aligned} &\sum_{\boldsymbol{S}\subseteq\mathcal{U}\setminus\{i,j\}} \frac{2}{(M - |\boldsymbol{S}| - 1)\binom{M}{|\boldsymbol{S}|+1}} \\ &= 2\sum_{|\boldsymbol{S}|=0}^{M-2} \frac{\binom{M-2}{|\boldsymbol{S}|}}{(M - |\boldsymbol{S}| - 1)\binom{M}{|\boldsymbol{S}|+1}} \\ &= 2\sum_{|\boldsymbol{S}|=0}^{M-2} \frac{(M - |\boldsymbol{S}| - 1)!(|\boldsymbol{S}| + 1)!(M-2)!}{(M - |\boldsymbol{S}| - 1)M!(M - |\boldsymbol{S}| - 2)!|\boldsymbol{S}|!} \\ &= \frac{2\sum_{|\boldsymbol{S}|=0}^{M-2}(|\boldsymbol{S}| + 1)}{M(M-1)} = 1. \end{aligned}$$

□

D. Proof of Remark 2

We prove Remark 2 in this section.

Proof. Remark 2 generalizes Theorem 2 to multiple features. Specifically, we consider a subset of features $\mathcal{V} = \{j_1, j_2, \cdots\}$. According to Equation (5), we have

$$\begin{aligned}
&\left|\phi_i(f_v, \mathcal{U}) - \phi_i(f_v, \mathcal{U} \setminus \mathcal{V})\right| \\
&= |\phi_i(f_v, \mathcal{U}) - \phi_i(f_v, \mathcal{U} \setminus \{j_1\}) \\
&\qquad + \phi_i(f_v, \mathcal{U} \setminus \{j_1\}) - \phi_i(f_v, \mathcal{U} \setminus \{j_1, j_2\}) \\
&\qquad + \cdots| \\
&\leq |\phi_i(f_v, \mathcal{U}) - \phi_i(f_v, \mathcal{U} \setminus \{j_1\})| \\
&\qquad + |\phi_i(f_v, \mathcal{U} \setminus \{j_1\}) - \phi_i(f_v, \mathcal{U} \setminus \{j_1, j_2\})| \\
&\qquad + \cdots \\
&= \sum_{j \in \mathcal{V}} \epsilon_{i,j} |x_i - \bar{x}_i||x_j - \bar{x}_j|
\end{aligned}$$

After taking $\mathcal{S} = \mathcal{U} \setminus \{i\} \setminus \mathcal{V}$, we have

$$\left|\phi_i(f_v, \mathcal{U}) - \phi_i(f_v, \mathcal{S} \cup \{i\})\right| \leq \sum_{j \in \mathcal{U} \setminus \{i\} \setminus \mathcal{S}} \epsilon_{i,j} |x_i - \bar{x}_i||x_j - \bar{x}_j|$$

□

E. Proof of Equation (9) to Equation (10)

We give the detailed proof from Equation (9) to Equation (10) in this section.

Proof. According to Equation (10), for $\mathcal{S} \subseteq \mathcal{U} \setminus \{i\}$, we have that

$$\sum_{j \in \mathcal{U} \setminus \mathcal{S}} \epsilon_{i,j} |x_i - \bar{x}_i||x_j - \bar{x}_j| \tag{34}$$

$$= \sum_{j \in \mathcal{U}} \epsilon_{i,j} |x_i - \bar{x}_i||x_j - \bar{x}_j| \tag{35}$$

$$- \sum_{j \in \mathcal{S}} \epsilon_{i,j} |x_i - \bar{x}_i||x_j - \bar{x}_j|, \tag{36}$$

where $\sum_{j \in \mathcal{U}} \epsilon_{i,j} |x_i - \bar{x}_i||x_j - \bar{x}_j|$ is a constant for the target feature i. In this way, Equation (10) can be considered as the dual problem of Equation (9). □

F. Details about the Datasets

We give the details about the datasets in this section, and the dataset statistics are shown in Table 2.

- **Census Income**[3]: In this dataset, each sample has five continuous features and eight one-hot encoded categorical features. The task for this dataset is to predict whether a person has income *more than* (*target*=1) or *less than* (*target*=0) 50K/yr based on her/his personal features (*education, occupation, working hours, etc.*).
- **German credit**[4]: The samples in this dataset have seven continuous features and nine one-hot encoded categorical features. The task is to predict whether a person has *good* (y=1) or *bad* (y=0) credit risks based on her/his personal features (*job, education, balance, etc.*).
- **Criteo**[5][6]: Each sample in this dataset corresponds to a user and an ad which have 13 continuous features and 26 one-hot encoded categorical features. The Criteo dataset is widely used in recommender systems, where the task is to predict the clicking rate of a user on an ad.

G. Details about the Baseline Methods

We give the details about the baseline methods in this section.

- **Kernel-SHAP** (Lundberg & Lee, 2017): Kernel-SHAP is a model agnostic method for feature importance estimation, which uses specific linear regressions to approach the original model outputs at each data point so that the linear weights approach the feature contribution. Kernel-SHAP estimates the feature contributions by

$$[\hat{\phi}_1, \cdots, \hat{\phi}_M] = \left(\mathbf{A}^\intercal \mathbf{W} \mathbf{A}\right)^{-1} \mathbf{A}^\intercal \mathbf{W} \mathbf{b}, \tag{37}$$

  where $\mathbf{A} = [\mathbf{1}_{\boldsymbol{S}_1}^\intercal, \cdots, \mathbf{1}_{\boldsymbol{S}_N}^\intercal]$; $\mathbf{W} = \text{diag}\{\pi_1, \cdots, \pi_N\}$; and for $1 \leq n \leq N$, π_n is given by

$$\pi_n = \frac{M - 1}{\binom{M}{|\boldsymbol{S}_n|} |\boldsymbol{S}_n| |M - |\boldsymbol{S}_n||}. \tag{38}$$

- **KS-WF** (Covert & Lee, 2021): KS-WF adopts the Welford algorithm (Welford, 1962) to calculate and reduce the variance of the feature contribution estimation. Specifically, in each iteration n, it randomly selects feature subset $\boldsymbol{S}_n$, and calculates the coefficient matrix by

$$\begin{aligned}
\mathbf{A}_n &= \left(1 - \frac{1}{n}\right) \mathbf{A}_{n-1} + \frac{1}{n} \mathbf{1}_{\boldsymbol{S}_n} \mathbf{1}_{\boldsymbol{S}_n}^\intercal \\
\mathbf{b}_n &= \left(1 - \frac{1}{n}\right) \mathbf{b}_{n-1} + \frac{1}{n} \Big(f_v(\mathcal{U}) - f_v(\varnothing)\Big) \mathbf{1}_{\boldsymbol{S}_n},
\end{aligned}$$

  where $\mathbf{A}_0 = \mathbf{0}_{M \times M}$ and $\boldsymbol{b}_0 = \mathbf{0}_M$ for the first iteration. After the n iterations, the feature contribution are estimated by

$$[\hat{\phi}_1, \cdots, \hat{\phi}_M] = \mathbf{A}_n^{-1} \Big(\mathbf{b}_n - \mathbf{1} \frac{\mathbf{1}^\intercal \mathbf{A}_n^{-1} \mathbf{b} - f_v(\mathcal{U}) + f_v(\varnothing)}{\mathbf{1}^\intercal \mathbf{A}_n^{-1} \mathbf{1}}\Big).$$

[3] https://archive.ics.uci.edu/ml/datasets/census+income

[4] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

[5] https://www.kaggle.com/c/criteo-display-ad-challenge/data

[6] https://labs.criteo.com/2014/02/kaggle-display-advertising-challenge-dataset/

Table 2: Dataset Statistics

Dataset	Continuous	Categorical	Training	Validation	Testing
Census Income	5	8	20838	5210	6513
German Credit	7	9	28934	7234	9043
Cretio	13	26	80000	10000	10000

The iterative process of KS-WF has to follow the feed-forward pipeline without parallelization which constrains its computational efficiency.

- **KS-Pair** (Covert & Lee, 2021): KS-Pair adopts the pair sampling to accelerate the convergence of Kernel-SHAP. Specifically, KS-Pair samples pairs of input feature coalitions $(\boldsymbol{S}_n, \mathcal{U} \setminus \boldsymbol{S}_n)|_{1 \leq n \leq \frac{N}{2}}$ and estimates the feature contribution following Equation (37).
- **PS** (Mitchell et al., 2021): The permutation sampling method estimates the feature contribution merely based on model inference. It randomly masks each feature for each data point and takes the average variety of model output as the feature importance. Given the model value function f_v and input feature coalition $\boldsymbol{S}_1, \cdots, \boldsymbol{S}_N$, the feature contribution is estimated given by

$$\hat{\phi}_i = \frac{1}{N} \sum_{j=1}^{N} f_v(\boldsymbol{S}_n \cup \{i\}) - f_v(\boldsymbol{S}_n). \tag{39}$$

- **APS** (Lomeli et al., 2019; Mitchell et al., 2021): APS adopts the antithetical sampling to reduce the variance of feature contribution estimation. To be concrete, half of the feature coalitions are randomly selected from the feature space, and the remaining feature coalitions takes $S_{N+1-n} = \mathcal{U} \setminus \boldsymbol{S}_n$ for $\frac{N}{2} < n \leq N$. The feature contribution is estimated following Equation (39).

H. Implementation Details

The experiment on each dataset follows the pipeline of *model training*: training the DNN model; *interpretation benchmark*: adopting the brute-force algorithm to calculate the exact Shapley value as the ground truth explanation for the evaluation; and *interpretation evaluation*: evaluating the interpretation methods. Each step is specified as follows.

Model Training: We adopt 3-layer MLP (multi-layer perceptron) as the classification model for the Census Income and German Credit datasets. To train the model, we adopt Adam optimizer with 10^{-3} learning rate to update the model parameters so that the cross-entropy loss can be minimized. Adopting early stopping based on the performance on the validation dataset to avoid overfitting, we achieve 84.7% and 89.8% accuracy on the Census Income and German Credit testing set, respectively. For the Cretio dataset, we use DeepFM (Guo et al., 2017) as the model and adopt Adam optimizer with 10^{-4} learning rate to update the parameters. Other settings are the same with that of Census Income dataset, and we achieve 71.09% accuracy on the testing dataset.

Table 3: Hyper-parameter setting for model training and Shapley value benchmark.

Dataset	Census Income	German Credit	Cretio
Model	3-layer MLP	3-layer MLP	DeepFM
Hidden dim.	64	64	32
Opt., LR	Adam,10^{-3}	Adam,10^{-3}	Adam,10^{-4}
Batch Size	256	256	256
$\text{Ref}_{\text{continuous}}$	Mean	Mean	Mean
$\text{Ref}_{\text{categorical}}$	Mean	Mean	Mode

Interpretation Benchmark: We adopt the brute-force algorithm to calculate the ground-truth Shapley value (GT-Shapley value) for the evaluation. Specifically, the GT-Shapley values are calculated according to Equation (3), where $f_v(\boldsymbol{S})$ is given by Equation (2); the reference value of continuous features takes the mean value for all datasets; and that of categorical features takes the mean value for the Census Income and German Credit dataset, and takes the mode for the Cretio dataset[7]. Other hyper-parameter settings are summarized in Table 3.

Interpretation Evaluation: SHEAR and baseline methods are employed to generate interpretations for the instances in the testing set. To evaluate the interpretation generated by different methods, we have five evaluation metrics given in Sections 5.2, 5.3 and 5.4, including two metrics taking the GT-Shapley value as the reference: the absolute estimation error and accuracy of feature importance ranking; two metrics evaluating the interpretation via model perturbation: Faithfulness and Monotonicity; and the algorithmic throughput to evaluate the running speed of generating the explanation. The evaluation metrics are calculated for each testing instance, and we report the mean and standard deviation of the metric value to illustrate the interpretation performance and variance of SHEAR and baseline methods.

In the evaluation step, we have to unify the experimental conditions over different interpretation mechanisms to achieve a fair comparison. In particular, SHEAR and Permutation-

[7]DeepFM works based on the Hash-index of categorical features. Instead of the mean value of Hash-index, we adopt the mode for the reference value of categorical features.

based methods (i.e., PS and APS) take NM times of model evaluation to estimate the contributions of M features, while Kernel-SHAP-based methods (i.e., Kernel-SHAP, KS-WF and KS-Pair) simply take N times with the merit of matrix operations. For unified the conditions, we should make sure Kernel-SHAP-based methods can also benefit from NM times of model evaluation. Note that setting NM times of model evaluation in a single batch would not work for Kernel-SHAP-based methods in practice due to the M^2-time-related memory cost and computation load. We choose to execute Kernel-SHAP-based methods by M loops in our experiments, and average the contribution values as the output. In this way, we control the model evaluations, overall memory cost and computation load equal for all involved methods.

I. Details about Computing Infrastructure

The details about our physical computing infrastructure for testing the algorithmic throughput are given in Table 4.

Table 4: Computing infrastructure for the experiments.

Device Attribute	Value
Computing infrastructure	CPU
CPU model	Apple M1
CPU number	1
Core number	8
Memory size	16GB

We have not used GPUs in our experiments because SHEAR and baseline methods mostly rely on DNN evaluation instead of the training of DNNs. Since the DNN model for the interpretation is controlled equal for all methods, the ranking of algorithmic throughput in Figures 4 (a)-(c) should be roughly consistent with the testing results on other types of equipment.