CoShap: A Scalable Coalition Growth Approach to Shapley Value Approximation

JINGXUAN HE*, Guangxi University, China and Harbin Institute of Technology, China
CHANGSHUO LIU†, National University of Singapore, Singapore
SHAOFENG CAI, National University of Singapore, Singapore
XIXIAN HAN, Harbin Institute of Technology, China
YANYAN SHEN, Shanghai Jiao Tong University, China
BENG CHIN OOI, Zhejiang University, China

The Shapley value provides a theoretically grounded framework for model interpretability by quantifying feature importance, with critical applications in data analytics and data valuation. However, its exact computation is often intractable due to an exponential complexity that scales with the number of features. This poses a significant scalability challenge for high-dimensional datasets common in modern data systems. Prevailing approximation methods, typically based on Monte Carlo sampling, suffer from severe inefficiencies rooted in their unstructured and redundant exploration of the coalition space. These limitations lead to unreliable estimates and hinder practical adoption in data-intensive applications.

This paper presents CoShap, a novel framework that delivers fast and reliable Shapley value approximation through systematic coalition growth. CoShap consists of a two-phase evaluation: (1) A layer-wise coalition growth phase that incrementally builds larger feature coalitions from smaller ones. Inspired by Pattern Growth, this structured approach drastically reduces redundant model evaluations by reusing previously computed coalition utilities. (2) A subsequent feature-wise evaluation phase that adaptively allocates more computational budget to high-variance features, specifically those with the most uncertain contributions, to enhance approximation reliability. To guarantee an (ϵ, δ)-approximation of the true Shapley values, CoShap introduces a dynamic budget allocation method that balances the evaluation budget across both phases. Experiments on diverse datasets show that CoShap significantly outperforms state-of-the-art methods, achieving up to 7.15× speedup in runtime and a 76.84% reduction in approximation error.

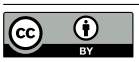
CCS Concepts: • **Computing methodologies** → **Feature selection**; • **Mathematics of computing** → **Statistical paradigms**.

Additional Key Words and Phrases: Model-agnostic interpretation, Shapley value approximation, AIxDB

ACM Reference Format:
Jingxuan He, Changshuo Liu, Shaofeng Cai, Xixian Han, Yanyan Shen, and Beng Chin Ooi. 2026. CoShap: A Scalable Coalition Growth Approach to Shapley Value Approximation. *Proc. ACM Manag. Data* 4, 3 (SIGMOD), Article 155 (June 2026), 27 pages. https://doi.org/10.1145/3802032

*Work done while at the National University of Singapore.
†Corresponding author.

Authors' Contact Information: Jingxuan He, Guangxi University, China, betterjingxuan@gmail.com and Harbin Institute of Technology, China, jingxuan@stu.hit.edu.cn; Changshuo Liu, National University of Singapore, Singapore, liu717@comp.nus.edu.sg; Shaofeng Cai, National University of Singapore, Singapore, shaofeng@comp.nus.edu.sg; Xixian Han, Harbin Institute of Technology, China, hanxx@hit.edu.cn; Yanyan Shen, Shanghai Jiao Tong University, China, shenyy@sjtu.edu.cn; Beng Chin Ooi, Zhejiang University, China, ooibc@zju.edu.cn.



ACM 2836-6573/2026/6-ART155
https://doi.org/10.1145/3802032

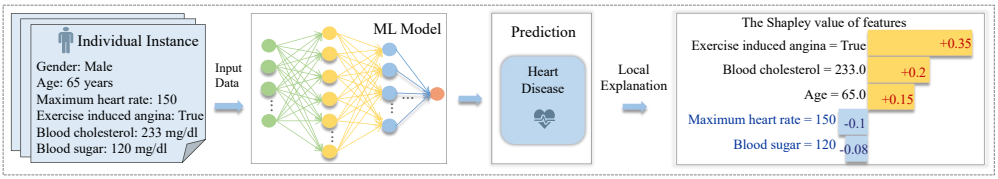


Fig. 1. Illustration of model interpretation via the Shapley value.

1 Introduction

In data-centric machine learning applications, particularly for high-stakes domains such as healthcare [3, 30] and finance [67], model interpretability is often as critical as predictive accuracy to foster trust and enable informed decision-making [2, 17, 28, 42]. Traditional interpretable models, such as linear regressions and decision trees [11], often lack the modeling capacity required for complex high-dimensional data. In contrast, while complex models such as neural networks [65] achieve superior predictive performance, their black-box nature can hinder practical adoption [63]. Model-agnostic explanation methods aim to bridge this divide [37, 44], with the Shapley value [7, 33, 61] emerging as a prominent approach due to its solid theoretical foundations. Beyond prediction explanation, the Shapley value has become a fundamental metric for quantifying data and participant contributions in core data management tasks, such as data repair [18], data valuation [24], data assemblage [33] and data sharing [60, 62].

The Shapley value [61], rooted in cooperative game theory, provides a principled means to fairly attribute the model prediction of a given instance to its input features by averaging their *marginal contributions* across all subsets of features [7]. For example, Figure 1 illustrates Shapley values explaining an individual prediction for a heart disease model. However, the exact Shapley value computation requires evaluating model predictions over all 2^n feature subsets or involves $O(n!)$ evaluations for naïve permutation-based methods [25], where n is the number of features. This exponential complexity renders exact computation intractable for models with even a moderate number of features, which poses a significant scalability challenge [32, 51].

In response to this computation bottleneck, recent studies have increasingly focused on developing Shapley value approximation (SVA) methods [11, 45, 50]. As shown in Algorithm 1, the standard Monte Carlo (MC) sampling approach [5] estimates Shapley values by averaging the marginal contributions from randomly sampled feature permutations, whose diverse feature orderings facilitate effective estimation. Fundamental to this process is the formation of feature subsets, called *coalitions*, and the evaluation of their model predictions, known as their *utilities*. While MC sampling and its variants, such as those employing stratified sampling [35], variance reduction techniques [4, 27], or antithetic and correlated samples [26, 64], offer feasible polynomial-time alternatives. They remain constrained by two fundamental limitations: (1) *computational inefficiency* stemming from the redundant evaluations of identical coalition utilities, and (2) *unreliable estimates* resulting from the neglect of the varying variances in different features' marginal contributions.

The inefficiency in existing SVA methods primarily stems from *redundant computation*. Specifically, in naive MC implementations, the utility $u(S)$ of a coalition S might be re-evaluated multiple times across different permutations. Moreover, when computing the marginal contribution $u(S \cup \{x_j\}) - u(S)$ for a feature x_j not in S, the utility $u(S)$ may be recomputed for each of the $n - |S|$ distinct features that could potentially be added to S. Although storing all the previously evaluated coalition utilities could theoretically eliminate such redundancies, this becomes infeasible due to the exponential number of possible coalitions. In practice, evaluating coalition utilities is computationally expensive, making redundant computations a major efficiency constraint. Table 1

Table 1. Computational redundancy in MC sampling.

Dataset	Number of Evaluations	Repeated Evaluations *	Re-evaluated Coalitions †
Bank Marketing ($n = 17$)	$100n$	343 (20.18%)	100 (7.37%)
	$500n$	2682 (31.55%)	665 (11.43%)
	$1000n$	6398 (37.64%)	1589 (14.99%)
	$5000n$	46349 (54.53%)	10586 (27.39%)
	$10000n$	108092 (63.58%)	23680 (38.25%)

∗ Evaluation Redundancy Rate: The percentage of duplicate subset evaluations, measuring the computational waste during sampling. Computed as: $\frac{\text{Repeated Evaluations}}{\text{Total Evaluations}} \times 100\%$.
† Coalitions Collision Rate: The percentage of unique coalitions that are evaluated more than once. Computed as: $\frac{\text{Re-evaluated Coalitions}}{\text{Total Unique Coalitions}} \times 100\%$.

quantifies the computational redundancy of the representative Monte Carlo (MC) method [5] on the Bank Marketing dataset. The results reveal an evaluation redundancy rate ranging from 20.18% to 63.58%. This redundancy is inherent to *stateless sampling*, where coalitions are generated independently without memory of prior evaluations. Consequently, a substantial portion of the computational budget is expended on re-computing identical utilities, creating a critical efficiency bottleneck.

Another critical limitation of current SVA methods is their failure to account for the heterogeneity among features, particularly in their marginal contribution variances [54]. Some features may exhibit low-variance marginal contributions, allowing for accurate Shapley value estimation with fewer permutation samples, whereas others demonstrate high variance, thus requiring more evaluations to achieve comparable approximation precision. Ignoring these variance differences causes a suboptimal allocation of the computation budget, which results in unnecessary evaluations on low-variance features, and more importantly, yields unreliable approximations for high-variance features due to insufficient sampling.

To address these limitations, we propose CoShap, a novel framework for fast and reliable Shapley value approximation based on systematic coalition growth. CoShap entails a two-phase evaluation: layer-wise evaluation and feature-wise evaluation. The first phase is a layer-wise coalition growth process that significantly reduces redundant computations by adopting principles from frequent pattern mining, akin to the FP-Growth method [23], which systematically builds larger feature coalitions from smaller ones, stratified by cardinality. This structured approach computes feature marginal contributions progressively, from smaller to larger coalitions, thereby eliminating the need to re-evaluate identical coalitions common in permutation-based sampling. Efficiency is further enhanced by reusing the utilities of shared coalitions in subsequent layer evaluations, in a way analogous to prefix reuse in frequent itemset mining. This allows computations to leverage coalition utilities evaluated and cached from preceding layers. The second phase is a feature-wise variance-aware allocation process, where CoShap addresses feature heterogeneity by adaptively allocating more computational budget to features exhibiting high evaluation variances. Higher variance in estimated marginal contributions indicates greater uncertainty, and thus, CoShap prioritizes these features by allocating them more coalition evaluations to refine their Shapley values. Such a strategy enhances estimation accuracy and reliability within a given computational budget. Furthermore, to provide an (ϵ, δ)-approximation of the true Shapley values, CoShap incorporates a variance-driven evaluation allocation method that dynamically balances the computation budget across both phases. With the two phases and dynamic budget allocation, CoShap delivers an efficient, accurate, and reliable SVA algorithm.

This work is part of NeurDB [39, 66], in which CoShap constitutes the explainable evaluation and decision-making layer. CoShap is employed to systematically quantify the contributions of individual features and actions. By providing principled and theoretically grounded attribution estimates, it enables the system to interpret how different inputs and operational choices influence downstream analytical outcomes. Consequently, CoShap facilitates interpretable data quality assessment and provides actionable insights that support informed decision-making within NeurDB.

The key contributions of this work include:

- We identify redundant computation and the neglect of inter-feature variance in existing SVA methods as primary causes of inefficiency and unreliability in existing SVA methods.
- We introduce a novel layer-wise coalition growth method that significantly reduces computation redundancy by adapting principles from frequent itemset mining.
- We develop a variance-aware evaluation method and a dynamic budget allocation method to enhance both efficiency and reliability by prioritizing high-variance features.
- We deliver a fast and reliable Shapley value approximation framework for model-agnostic interpretation. Experiments confirm the efficiency and accuracy of CoShap, achieving up to 7.15× speedup in runtime and 76.84% reduction in approximation error compared to state-of-the-art solutions.

The remainder of the paper is organized as follows. Section 2 introduces the background and preliminaries. Section 3 details the design of CoShap framework, followed by the theoretical analysis in Section 4. Section 5 presents the experimental results. Section 6 discusses the related work, and Section 7 concludes this paper.

2 Preliminaries

In this section, we introduce the preliminaries of Shapley value and its properties. We then define the problem of Shapley value approximation (SVA) and discuss the limitations of current research. Table 2 summarizes notations frequently used throughout the paper.

Shapley Value. The Shapley value, originating from cooperative game theory, fairly distributes collective gains among contributing players based on their individual contributions. In model interpretation, the Shapley value quantifies the contribution of each feature to a model's output, providing local interpretability by attributing the prediction for a specific instance to its constituent features [44].

Formally, given a set of n features $N = \{x_1, x_2, ..., x_n\}$, a subset $S \subseteq N$ is called a *coalition*. The *utility function*, $u(S)$, maps a coalition S to a real value, i.e., $u : 2^n \to \mathbb{R}$. In model explanation, the utility $u(S)$ typically represents the model's output when considering only the features in coalition S. The Shapley value ϕ_i of a feature $x_i \in N$ is then defined as:

$$\phi_i = \frac{1}{n} \sum_{S \subseteq N \setminus \{x_i\}} \frac{u(S \cup \{x_i\}) - u(S)}{\binom{n-1}{|S|}}, \tag{1}$$

where ϕ_i denotes the expected marginal contribution of feature x_i averaged over all possible coalitions $S \subseteq N \setminus \{x_i\}$. The *weighting factor*, equivalent to $\frac{1}{n} \cdot \frac{1}{\binom{n-1}{|S|}} = \frac{|S|!(n-|S|-1)!}{n!}$, ensures that each coalition S is considered equally in the computation of Shapley value, where the factor $\frac{1}{n}$ accounts for all n possible coalition sizes, i.e., *cardinalities* of coalitions, from 0 to n-1, and the binomial coefficient $\binom{n-1}{|S|}$ represents the number of coalitions with the same cardinality as S (without feature x_i).

Alternatively, Shapley values are more commonly computed by averaging marginal contributions over all possible feature permutations [59, 64]. For a feature set N and $\Pi(N)$ the set of all $n!$

Table 2. Frequently used notations.

Notation	Description
n	the number of model features
m	the number of model evaluations
N	a set of n features $N = \{x_1, x_2, ..., x_n\}$
S	a subset of features (called a *coalition*)
$u(S)$	the utility of the coalition S
$\Pi(N)$	the set of all possible permutations of features in N
π	a permutation in $\Pi(N)$
$\hat{\phi}_i$	the estimated Shapley value of feature x_i
μ_f	the proportion of budget for feature-wise evaluation
μ_n	the proportion of features for feature-wise evaluation

permutations of N, Shapley value can be computed by:

$$\phi_i = \frac{1}{|\Pi(N)|} \sum_{\pi \in \Pi(N)} [u(S_\pi^i \cup \{x_i\}) - u(S_\pi^i)], \tag{2}$$

where S_π^i is the set of features preceding x_i in a specific permutation $\pi \in \Pi(N)$. This essentially averages marginal contributions across all possible permutations for feature x_i.

Properties of Shapley Value. The Shapley value is widely recognized as the only measure that satisfies all the four desirable properties: efficiency, symmetry, linearity, and dummy [7].

- *Efficiency*. The sum of Shapley values for all features equals the total utility of the grand coalition N, i.e., $\sum_{i=1}^{n} \phi_i = u(N)$.
- *Symmetry*. If two features contribute equally to all coalitions, their Shapley values are identical, i.e., if for all $S \subseteq N \setminus \{x_i, x_j\}$, $u(S \cup \{x_i\}) = u(S \cup \{x_j\})$, then $\phi_i = \phi_j$.
- *Additivity*. For two games with utility functions u_1 and u_2, the Shapley value of a feature in the combined game is the sum of its values in each game, i.e., $\phi_i^{u_1+u_2} = \phi_i^{u_1} + \phi_i^{u_2}$.
- *Dummy*. A feature that has no impact on any coalition's utility has a Shapley value of zero, i.e., if for all $S \subseteq N \setminus \{x_i\}$, $u(S \cup \{x_i\}) - u(S) = 0$, then $\phi_i = 0$.

Local Interpretation using Shapley Value. In explainable machine learning, *local interpretation* refers to explaining a specific model prediction for an input instance $\mathbf{x}$ by quantifying the contributions of individual features [32]. Specifically, given a model f and an instance $\mathbf{x}$ with features $N = \{x_1, x_2, \ldots, x_n\}$, the utility of a coalition $S \subseteq N$ is defined as $u(S) = f(\mathbf{x}')$, where $\mathbf{x}'$ is the modified instance using features in S, while features not in S are replaced by corresponding reference values, e.g., the mean for numerical features or the mode for categorical features. This utility function measures the model prediction based on selected features in S. The Shapley value then quantifies the contribution of each feature $x_i \in \mathbf{x}$ by averaging its *marginal contributions*, i.e., $u(S \cup \{x_i\}) - u(S) = 0$, across all coalitions without x_i.

Shapley Value Approximation (SVA). While Shapley values possess desirable properties for local interpretation, their exact computation demands an exponential number of model evaluations, making it computationally prohibitive, particularly for models with many features or complex architectures. Consequently, Shapley value approximation methods are necessary in practice [4, 44]. Monte Carlo (MC) sampling [5] is a widely adopted approach for SVA. As detailed in Algorithm 1, MC randomly generates permutations of features and computes their marginal contributions (Lines 2-4). These contributions are then averaged to estimate the Shapley value (Line 8). While MC

Algorithm 1 The Monte Carlo sampling approach (MC)

Input: A set of features $N = \{x_1, x_2, ..., x_n\}$, the evaluation budget m.
Output: the estimated Shapley value $\hat{\phi}_i$ of each feature $x_i \in N$.

Initialize $\hat{\phi}_i = 0$ for $i = 1, ..., n$
for $j = 1$ to m **do**
  Generate a random permutation π_j of features
  for each feature x_i in π_j **do**
    $\hat{\phi}_i$ += $u(S^i_{\pi_j} \cup \{x_i\}) - u(S^i_{\pi_j})$
  end for
end for
for $i = 1$ to n **do**
  $\hat{\phi}_i = \frac{\hat{\phi}_i}{m}$
end for
return $\hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_n$.

sampling provides an unbiased estimate, this approach still requires a large number of model evaluations to achieve high accuracy, and therefore, more efficient approximation methods are much needed.

3 The CoShap Framework

In this section, we present CoShap in detail. We first provide an overview of CoShap framework in Section 3.1, followed by the descriptions of the layer-wise coalition growth in Section 3.2 and the variance-aware allocation in Section 3.3. Finally, we describe the dynamic budget allocation method and elaborate on the key steps of CoShap in Section 3.4.

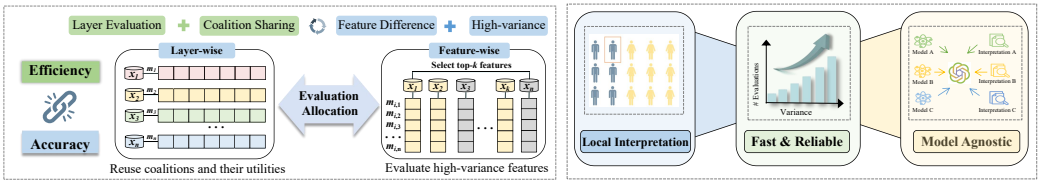
3.1 Overview

To achieve efficient and reliable SVA, we propose CoShap, a novel two-phase framework for model-agnostic interpretation. CoShap overcomes two critical limitations inherent in existing SVA methods to enhance efficiency and reliability.

The first phase, *layer-wise coalition growth*, directly addresses the issue of *collisions* and redundant model evaluations. In contrast to existing SVA methods that frequently re-sample and re-evaluate the same coalition, our method employs a pattern growth approach to construct coalitions incrementally. This provides a structural guarantee that each unique coalition S is generated and evaluated *at most once*. In this way, CoShap systematically eliminates redundant computations by reusing coalition utilities for a highly efficient approximation process.

The second phase, *feature-wise variance-aware allocation*, tackles the limitations of treating all features uniformly during evaluation. Recognizing the heterogeneity in feature contributions, we introduce a feature-wise variance-aware strategy that prioritizes features based on their evaluation variance. By allocating a larger proportion of the computational budget to features exhibiting high variance, this strategy significantly improves both the accuracy and reliability of the final approximation.

These two phases are integrated through a *dynamic budget allocation* mechanism. As illustrated in Figure 3, this strategy intelligently manages the trade-off between computational efficiency and estimation accuracy, ensuring an efficient and reliable Shapley value approximation.



(a) Overview of the CoShap framework. (b) Characteristics of CoShap.

Fig. 2. Overview of CoShap for local interpretation.

3.2 Layer-wise Coalition Growth

The inefficiency in existing SVA methods [5, 26, 64] mainly stems from computational redundancy, specifically the repeated evaluation of identical coalitions. CoShap addresses this issue through a layer-wise coalition growth process to incrementally construct coalitions stratified by cardinality. This approach allows for the systematic reuse of coalition utilities through common prefixes.

Given a set N of n features, the Shapley value computation can be structured into n layers, each corresponding to a specific coalition cardinality $j \in \{0, \ldots, n-1\}$:

$$\phi_i = \frac{1}{n}\sum_{j=0}^{n-1}\phi_{i,j} = \frac{1}{n}\sum_{j=0}^{n-1}\frac{1}{\binom{n-1}{j}}\sum_{S\subseteq N\setminus\{x_i\},|S|=j} u(S\cup\{x_i\}) - u(S), \tag{3}$$

where $\phi_{i,j}$ represents the Shapley value of feature $x_i \in N$ in layer L_j (coalitions of size j). In this phase, CoShap allocates an evaluation budget m_j to L_j and computes an estimated value $\hat{\phi}'_{i,j}$ as:

$$\hat{\phi}'_{i,j} = \frac{1}{m_j}\sum_{t=1}^{m_j} u(S_t\cup\{x_i\}) - u(S_t), \text{where } S_t \subseteq N\setminus\{x_i\}, |S_t| = j. \tag{4}$$

As illustrated in Figure 4 (left), CoShap computes the marginal contributions of features from low to high cardinality layers, avoiding redundant evaluation of the same coalitions across different permutations. To optimize memory utilization, CoShap only needs to maintain the coalition utilities $\{u(S) \mid S \in L_j\}$ necessary for computing marginal contributions in the subsequent layer L_{j+1}. Upon the completion of L_{j+1} evaluation, the memory allocated for L_j is immediately released. Further, CoShap computes the exact Shapley values for the first and last layers. Given that the two boundary layers contain only n coalitions each, their computation incurs a negligible linear cost. This eliminates sampling variance entirely for these two strata, thereby providing precise "anchors" for the estimation and yielding a significant accuracy improvement over purely random sampling.

To further enhance efficiency, as depicted in Figure 4 (right), CoShap reuses coalitions across layers. The coalition growth process constructs a new coalition S' by extending an existing one, S, with a feature $x_i \in N \setminus S$, i.e., $S' = S \cup \{x_i\}$. The marginal contribution of x_i can then be derived from the utility difference between S and S': $u(S') - u(S)$. Consequently, coalitions constructed within a layer are maintained and directly reused in the subsequent layers' computations. Crucially, a single coalition S from the current layer is reused up to $n - |S|$ times in the subsequent layer to compute marginal contributions.

Motivated by *pattern growth principles* [23], the proposed *structured layer-wise traversal* represents a fundamental shift from standard random sampling. By explicitly enabling the extensive reuse of coalition utilities, CoShap drastically reduces redundant model evaluations, thereby significantly enhancing computational efficiency.

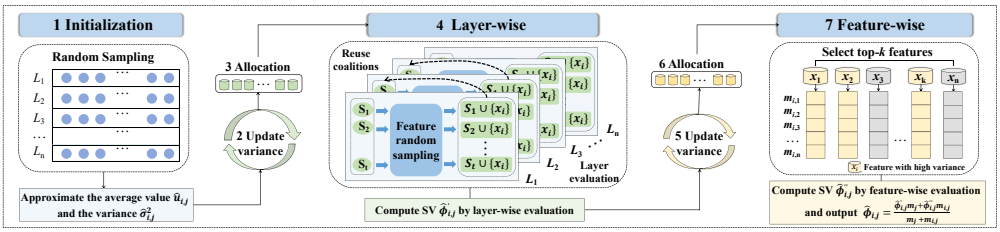


Fig. 3. Illustration of dynamic budget allocation in CoShap.

3.3 Feature-wise Variance-Aware Allocation

While the layer-wise coalition growth phase reduces redundant computations, it treats all features uniformly. This uniform treatment ignores feature heterogeneity, potentially yielding unreliable estimates for high-variance features whose marginal contributions are insufficiently captured by uniform sampling.

CoShap overcomes this by introducing a feature-wise variance-aware allocation process that prioritizes features with high estimation variances. This mechanism strategically allocates evaluation budget where it yields the most significant reduction in overall approximation error, thereby avoiding the inefficient expenditure of resources on well-estimated features. Based on the results of the coalition growth evaluation, CoShap identifies the top-k features with the highest evaluation variances, where $k = \lceil \mu_n \cdot |N| \rceil$, and μ_n specifies the proportion of features to be evaluated (sensitivity study provided in Section 5.4). Features with higher variance indicate greater uncertainty, thus demanding more evaluations for reliable estimates. Without loss of generality, let these features be indexed as $x_1, \ldots, x_k$, and $m_{i,j}$ be the number of evaluations for feature x_i in L_j during this phase, we estimate the Shapley value of x_i in L_j as:

$$\hat{\phi}_{i,j} = \frac{\hat{\phi}'_{i,j} m_j + \hat{\phi}''_{i,j} m_{i,j}}{m_j + m_{i,j}}, \tag{5}$$

where $\hat{\phi}''_{i,j} = \frac{1}{m_{i,j}} \sum_{t=1}^{m_{i,j}} u(S_t \cup \{x_i\}) - u(S_t)$ is the estimated Shapley value of x_i in L_j from this feature-wise evaluation. By directing more evaluations toward features with high variance, CoShap improves estimation accuracy and enhances computational efficiency through more targeted resource allocation. The process of determining m_j and $m_{i,j}$ is detailed in the following subsection.

3.4 Dynamic Budget Allocation

Existing evaluation allocation methods typically treat features uniformly, overlooking significant heterogeneity in contribution variances and thereby compromising both accuracy and reliability. To address this limitation, CoShap introduces a dynamic and variance-driven method that dynamically allocates the computational budget. This approach advances beyond conventional static sampling by strategically optimizing resource allocation to minimize estimation variance and enhance approximation reliability.

As illustrated in Figure 3, our dynamic budget allocation comprises three stages: (1) *Initial estimation* computes an initial set of variance estimates across all layers. (2) *Layer-wise allocation* then uses these estimates to guide the budget allocation for the coalition growth phase. (3) *Feature-wise allocation* refines estimates for a select group of top-k high-variance features identified in the prior stages. This structured allocation strategy is designed to provide an (ϵ, δ)-approximation guarantee on the final estimation error.

Algorithm 2 Approximation procedure of CoShap

Input: An instance $\mathbf{x}$ with features $N = \{x_1, x_2, ..., x_n\}$, the evaluation budget m, the initial estimation budget $m^* \geq 2$, $\mu_n, \mu_f \in (0, 1)$.

Output: the estimated Shapley value $\hat{\phi}_i$ of each feature $x_i \in N$.

$\hat{\phi}_{i,j} \leftarrow 0, \forall i \in \{1, 2, \ldots, n\}, \forall j \in \{0, 1, \ldots, n-1\}$
for $i = 1$ to n **do** △ Initial estimation
  for $j = 0$ to $n - 1$ **do**
    Sample m^* coalitions of size j that do not inclue feature i, compute the average marginal contribution $\hat{u}_{i,j}$ and its variance $\hat{\sigma}^2_{i,j}$
  end for
end for
$m = m - \sum_{i=1}^{n} \sum_{j=0}^{n-1} m^*$ // m is updated to the remaining budget
Compute the exact Shapley values for the first and the last layers
for $j = 1$ to $n - 2$ **do** △ Layer-wise evaluation
  Determine m_j as $m_j = \min(\frac{(\sum_{i=1}^{n} \hat{\sigma}^2_{i,j})^{1/2}(1-\mu_f)m}{\sum_{j=0}^{n-1}(\sum_{i=1}^{n} \hat{\sigma}^2_{i,j})^{1/2}}, \binom{n}{j})$, select m_j coalitions by sampling from the existing coalitions pool of cardinality j
  for $t = 1$ to m_j **do**
    For feature $x_i \notin S_t$, compute the marginal contribution as $u(S_t \cup \{x_i\}) - u(S_t)$
  end for
  Compute the average of marginal contributions $\hat{\phi}'_{i,j}$ in this step
end for
Update the variance $\hat{\sigma}_{i,j}$ and the remaining budget $m = m - \sum_{j=0}^{n-1} m_j$
Select $k = \lceil \mu_n \cdot |N| \rceil$ features with the highest evaluation variance
for each feature x_i in the top-k features **do** △ Feature-wise evaluation
  Determine $m_{i,j}$ as $m_{i,j} = \min(\frac{(\hat{\sigma}^2_{i,j})^{1/2}\mu_f m}{\sum_{i=1}^{k}(\sum_{j=0}^{n-1} \hat{\sigma}^2_{i,j})^{1/2}}, \binom{n-1}{j})$
  for $j = 1$ to $n - 2$ **do**
    Compute $\hat{\phi}''_{i,j} = \frac{1}{m_{i,j}} \sum_{t=1}^{m_{i,j}} u(S_t \cup \{x_i\}) - u(S_t)$
  end for
end for
for $i = 1$ to n **do**
  $\hat{\phi}_i = \frac{1}{n} \sum_{j=0}^{n-1} \hat{\phi}_{i,j} = \frac{1}{n} \sum_{j=0}^{n-1} \frac{\hat{\phi}'_{i,j} m_j + \hat{\phi}''_{i,j} m_{i,j}}{m_j + m_{i,j}}$
end for
return the estimated Shapley value of each feature $\hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_n$

Formally, given the evaluation budget m (the number of model evaluations), we formulate the allocation as an optimization problem to minimize the estimation error, defined as $\frac{1}{n} \sum_{i=1}^{n} |\hat{\phi}_i - \phi_i|$, the discrepancy between $\hat{\phi}_i$ and ϕ_i. We aim for an (ϵ, δ)-approximation, which ensures that the error is bounded by ϵ with a probability of at least $1 - \delta$, i.e., $\Pr\left(\frac{1}{n} \sum_{i=1}^{n} |\hat{\phi}_i - \phi_i| \leq \epsilon\right) \geq 1 - \delta$.

Initial Estimation. To guide the allocation, CoShap first conducts an initial estimate of the variance. This process is based on the analysis of variance principle (ANOVA) [48]. Specifically, CoShap performs m^* estimations (sensitivity study provided in Section 5.4) of marginal contributions for each feature within each layer to probe the variance distribution. Let $\hat{u}_{i,j} = \frac{1}{m^*} \sum_{t=1}^{m^*} (u(S_t \cup \{x_i\}) - u(S_t))$ be the average marginal contribution of feature x_i in layer L_j, where S_t is the t-th sampled coalition, $S_t \subseteq N \setminus \{x_i\}, |S_t| = j$. The corresponding estimated variance

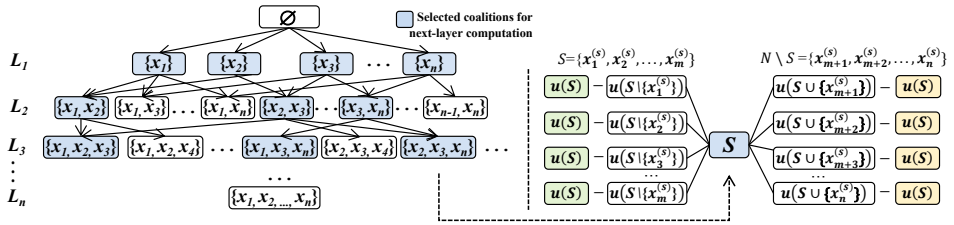


Fig. 4. The layer-wise evaluation and the reused coalitions.

$\hat{\sigma}_{i,j}^2$ can then be computed as: $\hat{\sigma}_{i,j}^2 = \frac{1}{m^*-1} \sum_{t=1}^{m^*} \left((u(S_t \cup \{x_i\}) - u(S_t)) - \hat{u}_{i,j}\right)^2$, where $m^* - 1$ is the Bessel correction term [20] ensuring that $E(\hat{\sigma}_{i,j}^2) = \sigma_{i,j}^2$.

This initial variance estimation provides the foundation for the subsequent allocation, which partition the remaining budget m into $m_L = (1 - \mu_f)m$ for layer-wise evaluation and $m_F = \mu_f m$ for feature-wise evaluation. To guarantee the (ϵ, δ)-approximation, the probability of the estimation error exceeding ϵ must be bounded by δ. As derived from concentration inequalities (see proof in Section 4.2), a sufficient condition to achieve this is:

$$\sum_{i=1}^{n} \sum_{j=0}^{n-1} \frac{\hat{\sigma}_{i,j}^2}{(1-\mu_f)m_j\epsilon^2} + \sum_{i=1}^{k} \sum_{j=0}^{n-1} \frac{\hat{\sigma}_{i,j}^2}{\mu_f m_{i,j}\epsilon^2} \leq \delta, \tag{6}$$

where the two terms on the left-hand side represent the contributions to the error probability bound from the layer-wise and feature-wise evaluations, respectively. The optimization for the subsequent budget allocation is therefore to distribute the budgets m_L and m_F to minimize each term in Eq. (6). Next, we detail the allocation method for each phase, including the derivation of the optimal distributions for m_j and $m_{i,j}$.

Layer-wise Allocation. The objective of layer-wise allocation is to minimize the first term $\sum_{i=1}^{n} \sum_{j=0}^{n-1} \frac{\hat{\sigma}_{i,j}^2}{(1-\mu_f)m_j\epsilon^2}$ in Eq. (6), subject to the budget constraint: $\sum_{j=0}^{n-1} m_j = m_L$. This constrained optimization can be solved via the Lagrangian function:

$$\mathcal{L}(m_j, \lambda) = \sum_{i=1}^{n} \sum_{j=0}^{n-1} \frac{\sigma_i^2}{m_L\epsilon^2} + \lambda \left(m_L - \sum_{j=0}^{n-1} m_j \right). \tag{7}$$

Taking the partial derivative with respect to each m_j and setting it to zero gives: $m_j = \frac{(\sum_{i=1}^{n} \hat{\sigma}_{i,j}^2)^{1/2}}{\sqrt{\lambda}}$, which shows that the optimal number of samples m_j for layer L_j is proportional to its standard deviation. By substituting this result back into the budget constraint, we obtain the solution for λ and the optimal allocation of the number of evaluations m_j in layer L_j:

$$m_j = \frac{(\sum_{i=1}^{n} \hat{\sigma}_{i,j}^2)^{1/2}}{\sum_{l=0}^{n-1} (\sum_{i=1}^{n} \hat{\sigma}_{i,l}^2)^{1/2}} \cdot (1 - \mu_f)m, \tag{8}$$

where $\hat{\sigma}_{i,j}^2$ is the variance from the initial estimation. This process ensures that layers with higher variance and thus greater uncertainty receive a proportionally larger share of the evaluation budget for efficient SVA.

Feature-wise Allocation. The objective of feature-wise allocation is to minimize the second term $\sum_{i=1}^{k} \sum_{j=0}^{n-1} \frac{\hat{\sigma}_{i,j}^2}{\mu_f m_{i,j}\epsilon^2}$ in Eq. (6), subject to the budget constraint: $\sum_{i=1}^{k} \sum_{j=0}^{n-1} m_{i,j} = m_F$. Before this allocation, we first dynamically update the variance estimates for the top-k features using

information pooled from both the initial (m^* samples) and layer-wise (m_j samples) stages via the combined variance formula:

$$\hat{\sigma}_{i,j}^2 = \frac{\hat{\sigma}_{i,j}^2 m^* + \hat{\sigma}_{i,j}'^2 m_j + \frac{m^* m_j}{m^* + m_j}(u_{i,j} - \phi_{i,j}')^2}{m^* + m_j}. \tag{9}$$

With the updated variances, we then select the top-k high-variance features to refine their Shapley value estimates for improved accuracy and reliability. Similarly, the optimal allocation for $m_{i,j}$ can be obtained by solving the constrained optimization via the Lagrangian function:

$$m_{i,j} = \frac{(\hat{\sigma}_{i,j}^2)^{1/2}}{\sum_{i=1}^{k}(\sum_{j=0}^{n-1}\hat{\sigma}_{i,j}^2)^{1/2}} \cdot \mu_f m, \tag{10}$$

where the denominator is the sum of standard deviations across all selected top-k features and all layers. This ensures that the evaluation budget is precisely targeted at the greatest sources of uncertainty, thereby improving the reliability of the final estimates.

We outline the key steps of CoShap in Algorithm 2. The procedure begins with an initial estimation process, where we compute the variance $\hat{\sigma}_{i,j}$ for each layer by randomly selecting m^* coalitions of features per layer (Lines 2-6). Then, we compute the exact Shapley values for the first and last layers (Line 8). Second, in the layer-wise coalition growth phase, a budget of m_j evaluations is allocated to each layer L_j (Line 10). For each selected coalition S_t, the marginal contributions of feature x_i in the current layer is computed by $u(S_t \cup \{x_i\}) - u(S_t)$ (Lines 11-13). Then CoShap updates the variance $\hat{\sigma}_{i,j}$ for each feature (Line 16). Third, in the feature-wise variance-aware allocation phase, CoShap selects top-k features with the highest variance for further evaluation. For each selected feature x_i, CoShap allocates $m_{i,j}$ evaluations (Line 19) and computes a refined estimate $\hat{\phi}_{i,j}''$ by sampling new coalitions in each layer (Lines 20-22). Finally, the final estimated Shapley value $\hat{\phi}_i$ for each feature x_i is calculated by combining the intermediate estimates from the *Layer-wise* and *Feature-wise* phases, $\hat{\phi}_{i,j}'$ and $\hat{\phi}_{i,j}''$, respectively (Line 25).

4 Theoretical Analysis

This section presents a theoretical analysis of CoShap, focusing on three aspects: unbiasedness, convergence and complexity.

4.1 Unbiasedness Analysis

The unbiasedness of SVA requires that the expected value of the estimated Shapley value $\hat{\phi}_i$ equals the true Shapley value ϕ_i for every feature $x_i \in N$, i.e. $E(\hat{\phi}_i) = \phi_i, \forall x_i \in N, 1 \le i \le n$.

Theorem 1. *Given a set of features $N = \{x_1, x_2, ..., x_n\}$, the CoShap algorithm (Algorithm 2) provides an unbiased estimate of the Shapley value for each feature, i.e., $E(\hat{\phi}_i) = \phi_i$.*

Proof. The proof first establishes that the layer-wise estimate $\hat{\phi}_{i,j}$ is an unbiased estimator of the true layer-wise Shapley value $\phi_{i,j}$, and then shows that the total estimate $\hat{\phi}_i$ is unbiased by the linearity of expectation.

In CoShap, the final estimate $\hat{\phi}_{i,j}$ for feature x_i in layer L_j is a weighted average of two estimators: $\hat{\phi}_{i,j}'$ from the layer-wise phase and $\hat{\phi}_{i,j}''$ from the feature-wise phase. Both estimators are derived from averaging the marginal contributions. The expectation of any such estimator takes the general form:

$$E[\hat{\phi}_{i,j}] = E\left[\frac{1}{m_{i,j}}\sum_{t=1}^{m_{i,j}}(u(S_t \cup x_i) - u(S_t))\right] = \frac{1}{m_{i,j}}\sum_{t=1}^{m_{i,j}} E[u(S_t \cup x_i) - u(S_t)], \tag{11}$$

where $m_{i,j}$ is the number of samples for feature x_i in layer L_j, and each sampled coalition $S_t \subseteq N \setminus \{x_i\}$ has cardinality $|S_t| = j$. The corresponding true layer-wise Shapley value $\phi_{i,j}$ for all S is the average marginal contribution over all possible coalitions of size j:

$$\phi_{i,j} = \frac{1}{\binom{n-1}{j}} \sum_{S \subseteq N\setminus\{x_i\}, |S|=j} \left(u(S \cup \{x_i\}) - u(S)\right). \tag{12}$$

Crucially, for any given layer L_j, the sampling process ensures that each coalition S_t is drawn uniformly at random from the set of all valid coalitions. Therefore, the expectation of a single randomly sampled marginal contribution is the true layer-wise Shapley value: $E[u(S_t \cup \{x_i\}) - u(S_t)] = \phi_{i,j}$. Substituting this back into Eq. (11) confirms that the estimator is unbiased for the layer-wise Shapley value estimate $\hat{\phi}_{i,j}$: $E[\hat{\phi}_{i,j}] = \frac{1}{m_{i,j}}\sum_{t=1}^{m_{i,j}} \phi_{i,j} = \phi_{i,j}$.

Since both $\hat{\phi}'_{i,j}$ and $\hat{\phi}''_{i,j}$ are unbiased estimators of $\hat{\phi}_{i,j}$, their weighted average in Eq. 5 is also an unbiased estimator of $\phi_{i,j}$:

$$E[\hat{\phi}_{i,j}] = E\left[\frac{\hat{\phi}'_{i,j}m_j + \hat{\phi}''_{i,j}m_{i,j}}{m_j + m_{i,j}}\right] = \frac{E[\hat{\phi}'_{i,j}]m_j + E[\hat{\phi}''_{i,j}]m_{i,j}}{m_j + m_{i,j}} = \frac{\phi_{i,j}m_j + \phi_{i,j}m_{i,j}}{m_j + m_{i,j}} = \phi_{i,j}. \tag{13}$$

The total estimated Shapley value $\hat{\phi}_i$ of x_i is the average of the layer-wise estimates: $\hat{\phi}_i = \frac{1}{n}\sum_{j=0}^{n-1}\hat{\phi}_{i,j}$. By the *linearity of expectation*, we obtain:

$$E[\hat{\phi}_i] = E[\frac{1}{n}\sum_{j=0}^{n-1}\hat{\phi}_{i,j}] = \frac{1}{n}\sum_{j=0}^{n-1}E[\hat{\phi}_{i,j}] = \frac{1}{n}\sum_{j=0}^{n-1}\phi_{i,j} = \phi_i. \tag{14}$$

Therefore, $\hat{\phi}_i$ is an unbiased estimator of ϕ_i for all features x_i. □

4.2 Convergence Analysis

This subsection presents the convergence property of CoShap by deriving bounds on the number of evaluations and the estimation error. First, we formalize the property using concentration inequalities with Theorem 2 establishing the foundation for the subsequent theorems on sample complexity in Theorem 3 and error bound in Theorem 4. Moreover, we analyze the time complexity of CoShap. We provide detailed proofs and derivations in Appendix A.

Using the variance estimates, CoShap allocates the number of evaluations. Let $\mu_f \in (0,1)$ be the proportion of budget for the feature-wise variance-aware allocation phase, $\mu_n \in (0,1)$ be the proportion of features selected for this phase, and $k = \lceil u_n \cdot n \rceil$ be the number of features for feature-wise evaluation. Then, the number of evaluations in the layer-wise coalition growth phase is $(1-\mu_f)m = \sum_{j=0}^{n-1} m_j$, and the number of evaluations in the variance-aware allocation phase is $\mu_f m = \sum_{i=1}^{n}\sum_{j=0}^{n-1} m_{i,j}$. The theorem 2 below provides the bound of an (ϵ, δ)-approximation,

Theorem 2. *Given a maximum estimation error $\epsilon > 0$, CoShap has the bound $\sum_{i=1}^{n}\sum_{j=0}^{n-1}\frac{\hat{\sigma}^2_{i,j}}{(1-\mu_f)m_j\epsilon^2} + \sum_{i=1}^{k}\sum_{j=0}^{n-1}\frac{\hat{\sigma}^2_{i,j}}{\mu_f m_{i,j}\epsilon^2} \le \delta$ under the condition $Pr(|\hat{\phi}_{i,j} - \phi_{i,j}| \le \epsilon) \ge 1-\beta$, where $\delta = 1-(1-\beta)^2$.*

Proof. Given the maximum estimation error ϵ, under the condition $|\hat{\phi}_{i,j} - \phi_{i,j}| \leq \epsilon$ with probability at least $1-\beta$, we have:

$$Pr(\frac{1}{n}\sum_{i=1}^{n}|\hat{\phi}_i - \phi_i| \leq \epsilon) \geq \prod_{i=1}^{n}\prod_{j=0}^{n-1} Pr(|\hat{\phi}_{i,j} - \phi_{i,j}| \leq \epsilon) \geq 1-\delta, \tag{15}$$

where ϕ_i and $\hat{\phi}_i$ are the exact and the estimated Shapley value of feature x_i, respectively. With $\delta = 1-(1-\beta)^2$, to satisfy $\prod_{i=k+1}^{n}\prod_{j=0}^{n-1}(1-\frac{\hat{\sigma}_{i,j}^2}{(1-\mu_f)m_j\epsilon^2})\cdot\prod_{i=1}^{k}\prod_{j=0}^{n-1}[(1-\frac{\hat{\sigma}_{i,j}^2}{\mu_f m_{i,j}\epsilon^2})\cdot(1-\frac{\hat{\sigma}_{i,j}^2}{(1-\mu_f)m_j\epsilon^2})] \geq 1-\delta$, we use *the logarithms property* and *Taylor series expansion*, yielding:

$$\begin{aligned}&\sum_{i=k+1}^{n}\sum_{j=0}^{n-1}\frac{\hat{\sigma}_{i,j}^2}{(1-\mu_f)m_j\epsilon^2} + \sum_{i=1}^{k}\sum_{j=0}^{n-1}(\frac{\hat{\sigma}_{i,j}^2}{\mu_f m_{i,j}\epsilon^2} + \frac{\hat{\sigma}_{i,j}^2}{(1-\mu_f)m_j\epsilon^2})\\ &= \sum_{i=1}^{n}\sum_{j=0}^{n-1}\frac{\hat{\sigma}_{i,j}^2}{(1-\mu_f)m_j\epsilon^2} + \sum_{i=1}^{k}\sum_{j=0}^{n-1}\frac{\hat{\sigma}_{i,j}^2}{\mu_f m_{i,j}\epsilon^2} \leq \delta.\end{aligned} \tag{16}$$

□

4.2.1 Evaluation number bound. To determine the lower bound of evaluations m, we apply Chebyshev's inequality.

Theorem 3. *Given a maximum estimation error $\epsilon > 0$ and a confidence level $(1-\delta)$, μ_f is the proportion of the budget for variance-aware allocation phase, the evaluation budget m in CoShap satisfies the following bound:*

$$m \geq \frac{n\hat{\sigma}^2(\mu_f^2 + k(1-\mu_f)^2)}{\delta(\epsilon\mu_f(1-\mu_f))^2}. \tag{17}$$

4.2.2 Error bound. Given an estimation error ϵ and a confidence level δ, we present the Theorem 4 regarding the precision of the estimates by a pair of parameters via the parameter pair (ϵ, δ).

Theorem 4. *For a feature $x_i \in N$, the estimation error of ϕ_i can be bounded by $\sqrt{\frac{n\hat{\sigma}_i^2}{\delta m}}$, where $\hat{\sigma}_i$ is the estimation variance of ϕ_i.*

4.3 Complexity Analysis

In CoShap, the time complexity is primarily dominated by the cost of model evaluations for evaluating coalition utilities. Let T_e denote the time required for a single utility evaluation, which can be treated as a constant factor. The total number of such evaluations m is determined by the desired (ϵ, δ)-approximation guarantee, which requires $O(\frac{n\hat{\sigma}^2}{\delta\epsilon^2})$ number of model evaluations to satisfy the error bound in Theorem 3. Hence, the overall time complexity is dominated by $O(mT_e)$. Besides model evaluations, the computational overhead includes: (1) The initial variance estimation has a complexity of $O(n^2m^*)$. (2) The subsequent allocation and aggregation involve loops over features and layers, with a complexity of $O(n^2 + kn)$. In practice, the total number of model evaluations m is significantly larger than n and m^*. Therefore, the runtime of CoShap is determined by the evaluation budget m, which can be configured in our algorithm to meet practical research constraints.

We compare the theoretical error bounds of CoShap with state-of-the-art approximation methods [59, 64]. Given the number of features n, an evaluation budget m, and a confidence level $(1-\delta)$, methods based on stratified sampling provide an error bound of $\sqrt{\frac{n\sigma^2}{m}\ln(\frac{n}{\delta})}$, where σ^2 is the variance of the true Shapley values. In contrast, CoShap establishes a tighter bound of $\sqrt{\frac{n\sigma^2}{\delta m}}$ by

dynamically optimizing the budget allocation, which effectively eliminates the logarithmic factor inherent in standard uniform bounds.

5 Experiments

In this section, we evaluate the performance of CoShap in terms of accuracy and efficiency in Section 5.2, convergence, scalability and interpretability in Section 5.3, with comparison to state-of-the-art SVA baselines. Moreover, we conduct the sensitivity study in Section 5.4, a case study in Section 5.5 and the ablation study in Section 5.6, respectively.

5.1 Experimental Setup

For a fair comparison, we follow experimental settings similar to prior works [4, 26, 64]. All experiments are conducted on a server with 48 Intel Xeon Silver 4214R CPUs @ 2.4GHz and 128GB of memory, with results reported as the average of 20 trials. The code of CoShap is available at https://anonymous.4open.science/r/CoShap.

5.1.1 Datasets. We employ two widely used benchmark datasets from cooperative game theory [4], one table discovery dataset, and two real-world datasets from high-stakes domains, namely finance and healthcare. These datasets are chosen to cover a broad spectrum of application scenarios. Multilayer perceptron models are employed as the utility functions.

Airport Game [29]. In the airport game, an airstrip that accommodates a given plane can also accommodate smaller planes at no extra cost. The Shapley value in this game represents a fair distribution of costs among different planes requiring various airstrip lengths. There are 100 planes, and each plane i has an associated cost c_i. The utility function in this game is given by:

$$u(S) = \max_{i \in S} c_i$$

where $\{c_1, \ldots, c_{100}\} = \{\underbrace{1, \ldots, 1}_{8}, \underbrace{2, \ldots, 2}_{12}, \underbrace{3, \ldots, 3}_{6}, \underbrace{4, \ldots, 4}_{14}, \underbrace{5, \ldots, 5}_{8}, \underbrace{6, \ldots, 6}_{9}, \underbrace{7, \ldots, 7}_{9}, \underbrace{8, \ldots, 8}_{13}, \underbrace{9, \ldots, 9}_{10}, \underbrace{10, \ldots, 10}_{10}\}$.

Voting Game [40]. This dataset models a non-symmetric voting game with 51 players under majority rule, quantifying voting power through Shapley values. The utility function is given as follows:

$$u(S) = \begin{cases} 1, if \sum_{i \in S} w_i > \frac{1}{2} \sum_{j \in N} w_i \\ 0, otherwise \end{cases}$$

where w_i is the vote weight for the player x_i, and the weights are $\{w_1, \ldots w_{51}\}$ = {45, 41, 27, 26, 26, 25, 21, 17, 17, 14, 13, 13, 12, 12, 12, 11, 10, 10, 10, 10, 9, 9, 9, 9, 8, 8, 7, 7, 7, 7, 6, 6, 6, 6, 5, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3}.

Bank Marketing [38]. This dataset originates from marketing campaigns of a Portuguese banking institution, containing 4521 samples and 17 features. The campaigns aimed to predict client subscriptions to term deposits, with binary outcomes labelled as 'yes' or 'no'.

Heart Disease [43]. The heart disease dataset contains 40 features and 246,022 instances, recording factors impacting heart diseases. It was collected from a U.S. cross-sectional telephone survey conducted by the Centers for Disease Control and Prevention as a major part of the Behavioral Risk Factor Surveillance System [6].

Table Discovery [14]. This dataset introduces a novel task named natural language–conditional table discovery, where a query table is paired with natural language descriptions to retrieve relevant candidate tables. Specifically, a table join dataset with a maximum of 33 table columns and 4,821

Table 3. Effectiveness evaluation of CoShap.

Dataset	#Evals.* / n	MC		KernelSHAP		CC		CCN		S-SVARM		Diff		CoShap	
		err_{avg}	err_{max}	err_{avg}	err_{max}	err_{avg}	err_{max}	err_{avg}	err_{max}	err_{avg}	err_{max}	err_{avg}	err_{max}	$err^{\dagger}_{avg}$	$err^{\dagger}_{max}$
Airport Game	100	0.727	3.323	0.709	0.965	1.040	12.027	**0.225**	2.036	0.289	2.701	0.551	6.554	**0.099** (↓56.05%)	0.683 (↓66.44%)
	500	0.484	1.658	0.409	0.950	0.232	3.074	**0.101**	0.961	0.116	1.127	0.263	3.090	**0.035** (↓65.62%)	0.375 (↓60.98%)
	1000	0.415	1.261	0.339	0.938	0.160	1.606	0.077	0.889	**0.076**	0.857	0.189	1.759	**0.032** (↓57.65%)	0.315 (↓63.17%)
	5000	0.146	0.620	0.105	0.533	0.065	0.666	**0.034**	0.334	0.036	0.349	0.063	0.788	**0.019** (↓45.67%)	0.191 (↓42.69%)
	10000	0.133	0.516	0.088	0.391	0.048	0.556	0.025	0.274	**0.024**	0.232	0.049	0.561	**0.012** (↓47.35%)	0.147 (↓36.45%)
Voting Game	100	0.996	5.889	0.940	2.613	0.713	4.755	**0.292**	1.304	0.645	2.962	0.809	4.290	**0.282** (↓3.21%)	1.094 (↓16.14%)
	500	0.610	3.228	0.591	2.562	0.226	1.084	**0.202**	0.892	0.251	1.135	0.437	1.075	**0.184** (↓9.01%)	0.714 (↓20.02%)
	1000	0.445	3.058	0.394	2.199	**0.157**	0.860	0.189	0.724	0.180	0.785	0.227	0.943	**0.128** (↓18.58%)	0.449 (↓38.08%)
	5000	0.120	0.474	0.085	1.184	0.069	0.360	**0.063**	0.304	0.079	0.357	0.068	0.526	**0.055** (↓12.69%)	0.239 (↓21.38%)
	10000	0.097	0.342	0.071	0.855	0.049	0.241	**0.042**	0.168	0.055	0.269	0.047	0.281	**0.038** (↓9.52%)	0.147 (↓12.50%)
Bank Marketing	100	1.839	16.874	1.611	12.470	1.497	10.122	1.784	13.168	2.528	17.560	**1.387**	9.730	**1.132** (↓18.39%)	8.982 (↓7.69%)
	500	1.280	9.154	1.093	10.145	0.832	5.305	0.821	4.976	2.009	13.067	**0.704**	4.678	**0.535** (↓24.01%)	4.225 (↓9.68%)
	1000	1.273	10.589	0.978	9.198	0.757	5.219	0.628	3.743	1.115	10.573	**0.584**	3.451	**0.426** (↓27.06%)	3.186 (↓7.68%)
	5000	0.429	3.396	0.417	2.769	0.412	2.666	0.372	1.941	0.423	3.244	**0.354**	1.681	**0.315** (↓11.02%)	1.867 (↑11.06%)
	10000	0.411	3.288	0.376	2.483	0.367	2.125	0.329	1.884	0.404	3.225	**0.303**	1.290	**0.290** (↓4.29%)	1.850 (↑43.41%)
Heart Disease	100	1.350	11.171	1.221	10.985	1.336	13.269	**1.004**	9.748	1.258	16.001	1.742	20.663	**0.740** (↓26.29%)	8.353(↓14.31%)
	500	0.764	6.969	0.759	6.574	0.521	5.365	**0.405**	4.399	0.554	7.099	0.861	10.651	**0.284** (↓29.88%)	2.984 (↓32.17%)
	1000	0.435	2.908	0.419	3.861	**0.306**	3.169	0.436	5.677	0.438	6.480	0.562	6.209	**0.204** (↓33.33%)	2.028 (↓36.01%)
	5000	0.238	3.301	0.208	2.311	0.201	2.270	0.192	2.048	**0.189**	2.011	0.248	2.465	**0.084** (↓55.56%)	0.855 (↓57.48%)
	10000	0.240	3.321	0.154	1.832	**0.139**	1.445	0.144	1.903	0.152	1.798	0.163	1.448	**0.070** (↓49.64%)	0.724 (↓49.90%)
Table Discovery	100	0.764	1.674	0.516	2.413	**0.417**	1.871	0.480	2.313	0.510	2.391	0.551	2.695	**0.464** (↑11.27%)	1.662 (↓0.72%)
	500	0.649	1.423	0.289	1.451	**0.195**	0.919	0.242	1.196	0.216	1.052	0.265	1.271	**0.194** (↓0.51%)	0.832 (↓9.47%)
	1000	0.493	1.628	0.169	0.932	**0.124**	0.566	0.148	0.744	0.158	0.849	0.185	1.038	**0.115** (↓7.26%)	0.559 (↓1.24%)
	5000	0.136	0.420	0.076	0.375	0.073	0.350	**0.070**	0.322	0.075	0.389	0.089	0.450	**0.057** (↓18.57%)	0.184 (↓42.86%)
	10000	0.128	0.357	0.055	0.257	0.056	0.284	**0.052**	0.221	0.054	0.270	0.065	0.314	**0.049** (↓5.77%)	0.207 (↓6.33%)
Data Repair	100	0.248	0.999	**0.130**	0.682	0.333	2.750	0.299	2.416	0.310	2.312	0.341	2.888	**0.087** (↓33.08%)	0.325 (↓52.35%)
	500	0.214	0.682	**0.108**	0.408	0.145	1.140	0.120	0.830	0.224	1.619	0.168	1.441	**0.033** (↓69.44%)	0.139 (↓65.93%)
	1000	0.169	0.526	0.098	0.291	0.117	1.017	**0.095**	0.740	0.190	1.251	0.109	0.876	**0.022** (↓76.84%)	0.107 (↓63.23%)
	5000	0.064	0.364	0.092	0.212	0.046	0.325	0.051	0.473	0.186	1.290	**0.034**	0.153	**0.010** (↓70.59%)	0.053 (↓65.36%)
	10000	0.032	0.158	0.089	0.212	**0.024**	0.144	0.026	0.166	0.121	0.984	0.028	0.148	**0.007** (↓70.83%)	0.041 (↓71.53%)

∗ The table reports the normalized number of evaluations per feature, i.e., the total number of evaluations divided by the number of features n.

† Error reduction of CoShap over the best-performing baselines for err_{avg} (**bold**) and err_{max} (underlined).

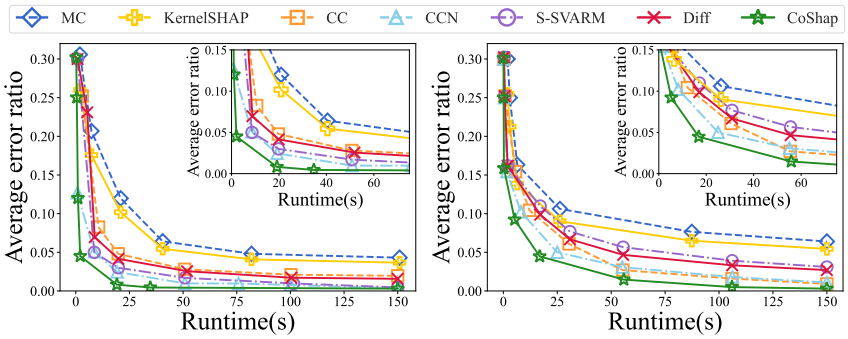
samples is utilized to identify candidate tables semantically or structurally related to the given query table.

Data Repair [46]. We utilize the Hospital dataset from HoloClean, collected from the U.S. Department of Health & Human Services and consists of 1,000 tuples with 19 attributes. The dataset is annotated with 9 real-world denial constraints, resulting in 6,604 constraint violations. Given an input tuple, our task is to predict whether it participates in any violations and to further analyze the influence of candidate repair actions on the violation outcomes.

5.1.2 Baselines. We compare CoShap with six baselines, categorized into *permutation-based* and *coalition-based* approaches. For *permutation-based* methods, we include the representative and classical Monte Carlo (MC) method, alongside the state-of-the-art CC and CCN algorithms. For *coalition-based* methods, we select the widely adopted KernelSHAP, as well as the recent advanced methods S-SVARM and Diff. All chosen baselines are model-agnostic, providing a direct and fair comparison to our proposed method.

MC [5] randomly samples permutations of features, and computes the marginal contribution of each feature. In the experiments, the Monte Carlo method serves as a benchmark for approximating the Shapley value by sampling a substantial number of permutations.

KernelSHAP [32] approximates Shapley values by constructing a weighted local surrogate model. It works by sampling feature coalitions and fitting a weighted least squares regression model, where the Shapley kernel defines the coalition weights. The regression coefficients from this fit are then interpreted as the estimated feature attributions for the given prediction.



(a) Airport Game (b) Voting Game

Fig. 5. Trade-offs between efficiency and accuracy.

CC [64] reformulates Shapley value estimation by introducing the complementary contributions. Instead of calculating the marginal contribution of each feature individually, CC estimates it by using the utilities of coalitions and their complements.
CCN [64] optimizes the sampling process of the CC method using the Neyman allocation. It divides complementary contributions into groups based on the cardinalities of coalitions, and then allocates samples to each group according to its variance.
S-SVARM [26] approximates the Shapley value by solving a regression problem constructed from randomly sampled coalitions and their corresponding model outputs, avoiding the direct computation of marginal contributions.
Diff [41] estimates the Shapley value by leveraging a differential matrix. It reduces the estimation variance by estimating the differences between Shapley values and employs a least squares optimization approach to estimate Shapley values from the matrix.

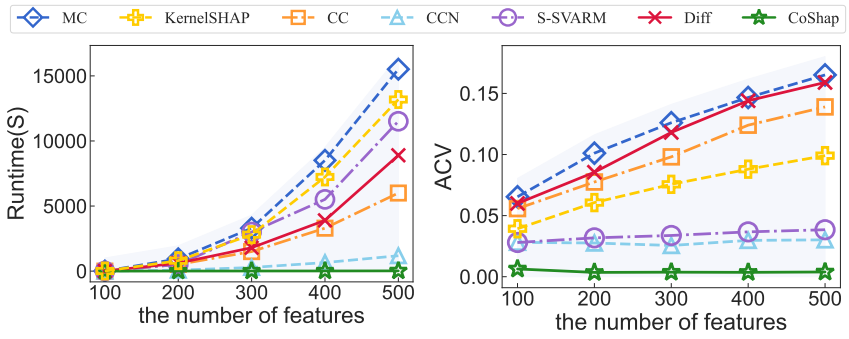
5.1.3 Evaluation Metrics. Our experiment utilizes three metrics to measure the performance of CoShap, the average error ratio (err_{avg}), the maximum error ratio (err_{max}) and the average coefficient of variation (ACV). The lower err_{avg}, err_{max} and ACV indicate the better quality of the estimated Shapley value.
The average error ratio reflects the ratio of the total accumulated absolute errors against the total Shapley value of features. Given the benchmark Shapley value ϕ_i and the approximate Shapley value $\hat{\phi}_i$, the err_{avg} is calculated as:

$$err_{avg} = \frac{1}{n}\sum_{i=1}^{n}|\frac{\hat{\phi}_i - \phi_i}{\phi_i}|, (1 \le i \le n). \tag{18}$$

The maximum error ratio provides insights into the worst-case scenario, ensuring the output interpretation is reliable. It is calculated as follows:

$$err_{max} = \max_i|\frac{\hat{\phi}_i - \phi_i}{\phi_i}|, (1 \le i \le n). \tag{19}$$

The Average coefficient of variation quantifies the relative dispersion of repeated Shapley value estimates. For feature x_i, let $\{\hat{\phi}_i^1, ..., \hat{\phi}_i^t\}$ be t independent estimates obtained under identical



(a) Runtime (b) ACV

Fig. 6. Scalability evaluation.

experimental setting, the ACV across all features is calculated as:

$$ACV = \frac{1}{n}\sum_{i=1}^{n} \frac{\sqrt{\frac{1}{t}\sum_{i=1}^{t}(\hat{\phi}_i^t - \frac{1}{t}\sum_{i=1}^{t}\hat{\phi}_i^t)^2}}{|\frac{1}{t}\sum_{i=1}^{t}\hat{\phi}_i^t|}, (1 \le i \le n). \tag{20}$$

5.1.4 Benchmark. Computing the exact Shapley value ϕ_i is impractical due to its exponential complexity in the number of features. Therefore, we use the true Shapley value reported in [4] as the benchmark for game theory datasets, and use the estimated Shapley value computed by the Monte Carlo method with $50000n$ permutations as the benchmark Shapley value for other experiments.

5.1.5 Parameters. The parameters for the evaluations are set as: m^*=max(2, $\lfloor \frac{m}{2n^2} \rfloor$), μ_f=0.1, and μ_n=0.5. This configuration is determined through a one-time grid search performed on a representative Airport Game dataset and consistently held fixed across all subsequent experiments. The computational overhead of this one-time search is negligible (under one minute and easily parallelized) and is therefore excluded from the runtime measurements reported in the experiments.

5.2 Accuracy and Efficiency

In this section, we evaluate the accuracy of CoShap in terms of err_{avg} and err_{max} by varying the number of evaluations m on five datasets. For efficiency, we evaluate the runtime to achieve target err_{avg} of CoShap and baselines on the five datasets.

Accuracy. As depicted in Table 3, CoShap outperforms all baselines. As m increases, the err_{avg} decreases, indicating that the approximated Shapley value becomes closer to the accurate Shapley value. Notably, in the data repair task, CoShap achieves significantly lower error ratios, showing its effectiveness in data management tasks. We observe that features with smaller Shapley values usually have larger error ratios. This is because even when the absolute error value is small, it can result in a large err_{avg} on a small Shapley value. CoShap achieves a small err_{avg} by strategically allocating more evaluations to high-variance features, thereby reducing estimation uncertainty. A similar trend is observed for err_{max}, which decreases for all methods as m increases, with CoShap consistently achieving the lowest error. This advantage is particularly pronounced when compared to kernelSHAP. While kernelSHAP remains competitive in terms of err_{max} on Data Repair under small evaluation budgets, it suffer from slow convergence on datasets with a larger feature space, e.g., Airport Game. In such high-dimensional settings, CoShap achieves substantially lower err_{avg}

Table 4. Efficiency evaluation of CoShap.

Dataset	err_{avg}	MC	KernelSHAP	CC	CCN	S-SVARM	Diff	CoShap †
Airport Game	≤**0.01**	–*	–	39.22s	20.87s	**9.40s**	47.41s	**3.13s** (↑3.00×)
	≤**0.05**	–	–	10.44s	8.62s	**6.41s**	10.68s	**1.31s** (↑4.89×)
	≤**0.1**	19.11s	13.11s	5.20s	**2.18s**	3.93s	5.04s	**1.01s** (↑2.16×)
Voting Game	≤**0.01**	–	–	30.72s	**19.76s**	39.76s	27.31s	**3.14s** (↑6.29×)
	≤**0.05**	922.15s	701.89s	15.71s	**5.69s**	22.70s	16.46s	**2.53s** (↑2.25×)
	≤**0.1**	3.44s	3.36s	3.25s	**2.40s**	3.71s	4.13s	**1.14s** (↑2.11×)
Bank Marketing	≤**0.1**	–	–	9882.81s	–	–	**5154.91s**	**4242.18s** (↑1.22×)
	≤**0.3**	–	2891.75s	2109.88s	3999.41s	1691.08s	**1293.89s**	**526.67s** (↑2.46×)
	≤**0.5**	716.72s	487.39s	367.23s	411.26s	299.76s	**243.74s**	**131.39s** (↑1.86×)
Heart Disease	≤**0.1**	–	–	6874.36s	**5953.66s**	–	–	**1649.61s** (↑3.61×)
	≤**0.3**	850.62s	835.19s	**637.15s**	760.59s	811.73s	1138.37s	**306.03s** (↑2.08×)
	≤**0.5**	432.12s	425.89s	**239.84s**	253.58s	365.05s	515.09s	**133.96s** (↑1.79×)
Table Discovery	≤**0.1**	–	691.42s	**569.89s**	607.36s	681.11s	705.09s	**196.10s** (↑2.91×)
	≤**0.3**	310.66s	134.26s	**97.81s**	105.90s	111.03s	125.30s	**62.71s** (↑1.56×)
	≤**0.5**	121.77s	70.59s	**51.33s**	59.98s	67.18s	78.34s	**29.61s** (↑1.73×)
Data Repair	≤**0.05**	2618.34s	–	712.67s	789.47s	–	**506.73s**	**98.43s** (↑5.15×)
	≤**0.1**	241.29s	**70.36s**	138.96s	125.46s	1042.38s	130.26s	**9.84s** (↑7.15×)
	≤**0.3**	25.89s	**9.76s**	16.56s	13.94s	14.26s	17.82s	**5.84s** (↑1.67×)

∗ Results marked with "–" denote that the runtime exceeded 10^4 seconds.

† Speedup of CoShap over the best-performing baselines for achieving a target err_{avg} (**bold**).

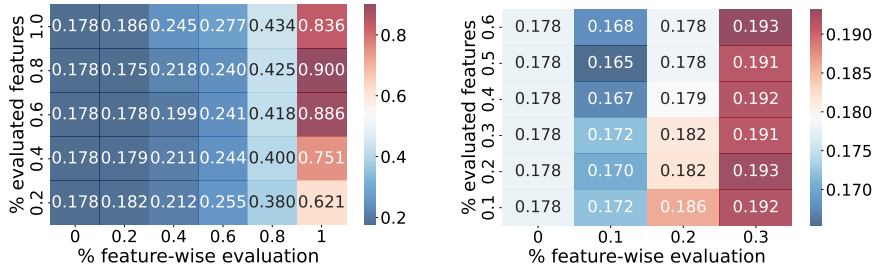
and err_{max}. This superior performance stems from CoShap's dynamical budget allocation, which actively directs computational resources toward high-variance features to refine their estimates more efficiently.

Efficiency. In Table 4, we investigate the runtime for the methods to achieve an average error ratio of ≤ 0.01, ≤ 0.05 and ≤ 0.1. The time cost by the baselines increases sharply as the number of features, while the proposed CoShap requires significantly less runtime to achieve the same approximation error ratio, which verifies the efficiency of our CoShap framework. Compared to these baselines, CoShap achieves a speedup over the best-performing baseline of 4.89×, 3.11×, 3.57×, 3.61×, 2.91× and 7.15× on Airport Game, Voting Game, Bank Marketing, Heart Disease, Table Discovery, and Data Repair datasets, respectively. There are three main reasons for such superior performance. First, CoShap reduces the time overheads by reusing the coalitions and their utility in the layer-wise coalition growth phase, therefore reducing the runtime of coalition calculations. Second, the feature-wise variance-aware allocation is adopted to achieve more efficient evaluation of feature contributions. Last, CoShap enhances efficiency by dynamically allocating evaluations based on estimated variance at each iteration, thus converging to the same estimated error level more effectively.

5.3 Convergence, Scalability & Interpretability

In this study, we assess the performance of CoShap in terms of convergence, scalability and interpretability, respectively.

Convergence. We evaluate convergence through error-runtime trade-off curves of methods across different datasets. These curves plot err_{avg} (horizontal axis) against runtime (vertical axis). In Figure 5(a), CoShap achieves the most rapid convergence rate, reducing errors 1.65× faster than baselines on Airport Game. It stabilizes at the error bound (0.005), outperforming baselines in precision. This indicates that CoShap achieves both rapid convergence and precise estimation. In Figure 5(b), CoShap maintains the fastest convergence rate and tighter error bound against all baselines. These results confirm that CoShap achieves an exponentially decreasing estimation error, aligning with the theoretical analysis in Section 4.



(a) Vary $\mu_f : 0 \rightarrow 1.0$ and $\mu_n : 0.2 \rightarrow 1.0$. (b) Vary $\mu_f : 0 \rightarrow 0.3$ and $\mu_n : 0.1 \rightarrow 0.6$.

Fig. 7. Effects of different parameter settings.

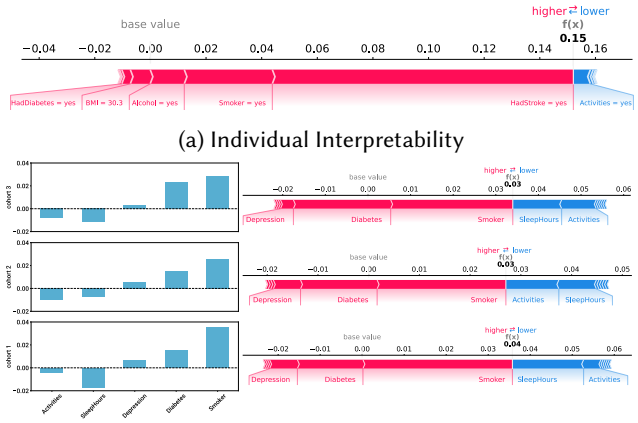
Scalability. We evaluate the scalability in terms of runtime and the *average coefficient of variation* (ACV) by simulating the Airport Game with 100, 200, 300, 400 and 500 features. Given a set of estimated Shapley values $\{\hat{\phi}_i^1, \ldots, \hat{\phi}_i^t\}$ computed t times under identical settings (where $\hat{\phi}_i^t$ denotes the t-th estimate for feature x_i), we evaluate ACV of CoShap and all baselines. As depicted in Figure 6(a), as the number of features increases from 100 to 500, the runtime of CoShap grows nearly linearly. In contrast, MC, KernelSHAP, S-SVARM, CC and Diff exhibit a quadratic or exponential increase in runtime, and CCN exhibits a slower runtime increase. Moreover, the ACV of CoShap remains stable as the number of features increases, indicating its scalability in handling high-dimensional data. The results reveal that CoShap achieves near-linear scaling behavior, which distinguishes CoShap from baselines.

Interpretability. We evaluate interpretability at both individual and cohort levels. At the individual level, we analyze feature contributions for heart disease prediction of a patient using Shapley values. As depicted in Figure 8(a), *Stroke* is identified as a primary factor, followed by *Smoker*, *Alcohol*, *BMI (Obesity)*, *Activity*, and *HadDiabetes*, which is consistent with previous population-attributable risk findings reported by [49]. Cohort-level [19] interpretation offers a broader and more stable perspective across groups of patients. Three distinct patient cohorts are derived based on age: *under 50*, *50 to 60*, and *over 60*. In Figure 8(b), *Smoker* consistently emerges as a primary risk factor for heart disease across all cohorts, while *Diabetes* contributes most substantially in cohort 3. *SleepHours* is more influential in cohort 1, whereas *Activity* is more significant in cohort 2. Furthermore, *Depression* has the minimal influence in cohort 3. These findings align with the results presented in [57].

5.4 Sensitivity Study

The Sensitivity Study experiment includes two aspects. Initially, we evaluate the proportion of budget μ_f and the proportion of features μ_n for feature-wise evaluation. Subsequently, we also evaluate the impact of the number of evaluations.

First, we focus on the two critical hyperparameters: μ_f and μ_n. These hyperparameters control the convergence of estimate error and the frequency of variance updates. As an illustration, the experimental results on Airport Game dataset in Figure 7 suggest that the optimal settings for μ_f lie between 0.1 and 0.2, and for μ_n between 0.4 and 0.5. In the absence of prior knowledge about a dataset, we recommend these default parameters as initial values, followed by a grid search to find better hyperparameters. Further, we evaluate the impact of the number of evaluations m^* in the initial estimation. The results in Figure 10 show that an excessively large m^* causes more accurate variance estimations, which can somewhat ensure a certain level of performance but at the expense of efficiency. Conversely, an overly small m^* reduces the time of initial estimation, but results in an



(a) Individual Interpretability

(b) Cohort Interpretability

Fig. 8. Interpretability analysis of CoShap.

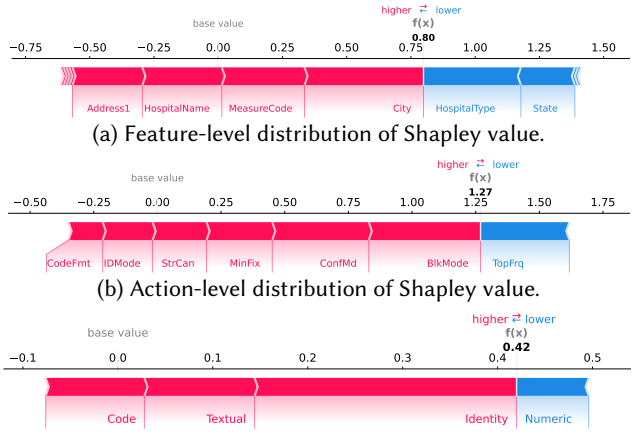
ineffective sample distribution, leading to a decline in evaluation performance. We recommend $m^* = \max(2, \lfloor \frac{m}{2n^2} \rfloor)$ to balance computational efficiency and estimation accuracy.

5.5 Case Study

To highlight the practical impact of CoShap in real-world applications, we conduct a case study on the data repair task, with results demonstrated in Figure 9. By formulating data repair as a cooperative game under integrity constraints, CoShap decomposes repair effectiveness through Shapley values and attributes the global repair outcome to individual features, repair actions, and heterogeneous feature groups.

Contribution of Important Features. CoShap is applied to calculate the Shapley value of individual features to identify those that drive the model's decisions. Figure 9 (a) reveals that attributes such as City and MeasureCode exhibit positive contributions, indicating that repairs in these features are more effective for violation correction. In contrast, HospitalType and State show negative contributions, suggesting that modifying these attributes alone is insufficient to resolve the underlying violations and may even increase the model's suspicion. This phenomenon is consistent with their limited involvement in the constraints, as each participates in only a single constraint, limiting their ability to provide corroborating evidence for violation correction.

Contribution of Specific Repairs. Following common data repair practices in the data cleaning literature [12, 15], we investigate seven representative repair strategies that capture a diverse spectrum of data repair actions and CoShap is employed to quantify their effectiveness. **Block Mode (BlkMode)** enforces consistency by replacing conflicting values within an equivalence block with the majority value. **Consensus-confident Mode (ConfMd)** applies a similar strategy but only when the majority value exceeds a confidence threshold. **Minimal Fix (MinFix)** repairs target only minority tuples within a block, preserving dominant values while minimizing intervention. **String Canonicalization (StrCan)** normalizes textual attributes by removing superficial variations such as case, punctuation, and spacing. **Anchor-strong ID Mode (IDMode)** performs consistency repair within sub-blocks defined by strong identifiers. **Code Format (CodeFmt)** corrects structured attributes with predefined formats, such as phone numbers and zip codes. Finally, **Top-3 Frequent Substitution (TopFrq)** heuristically replaces a value with one of the most frequent candidates in the block. Figure 9 (b) shows that consistency-enforcing repairs, such as BlkMode and ConfMd, contribute disproportionately to violation repairing, despite generating fewer candidate actions,



(a) Feature-level distribution of Shapley value.

(b) Action-level distribution of Shapley value.

(c) Group-level distribution of Shapley value.

Fig. 9. Shapley-based analysis of data repair tasks.

whereas frequency-based heuristics produce a large repair space with limited effectiveness. This observation aligns with the construction of the HoloClean hospital dataset, which introduces violations by corrupting a limited number of cells under predefined integrity constraints. Within each equivalence block, the underlying ground truth values are largely fixed and well-defined. Consequently, consistency-based repair actions constitute reasonable and semantically grounded repair candidates.

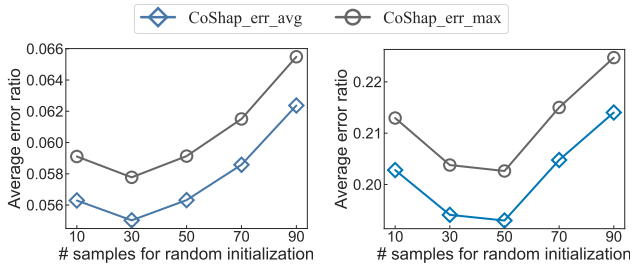
Influence of Heterogeneous Features. CoShap is leveraged to analyze the influence of different feature groups. We categorize features into four mutually exclusive groups based on their semantic roles [46]: identity features (e.g., HospitalName, HospitalOwner), textual features describing entities, code features with structured formats (e.g., PhoneNumber, ZipCode), and numeric features (e.g., Score, Sample). Figure 9 (c) demonstrates a clear hierarchy among heterogeneous features. Repairs on identity features are most effective in resolving violations. Textual and code features contribute positively but primarily play refinement-oriented roles. In contrast, numeric features exhibit negative contributions, suggesting that quantitative repairs are often ineffective. This effect arises because numeric values within an equivalence block naturally vary with the state-level aggregate statistic (Stateavg), making local numeric corrections insufficient to resolve underlying inconsistencies.

In summary, CoShap prioritizes high-impact features and repair actions while identifying low-utility or even detrimental repair directions, thereby effectively reducing the search space of candidate repairs. These findings position CoShap as a practical decision-support tool for data repair systems, enabling principled repair design, pruning of low-yield candidates, and more effective allocation of repair budgets in data management tasks.

5.6 Ablation Study

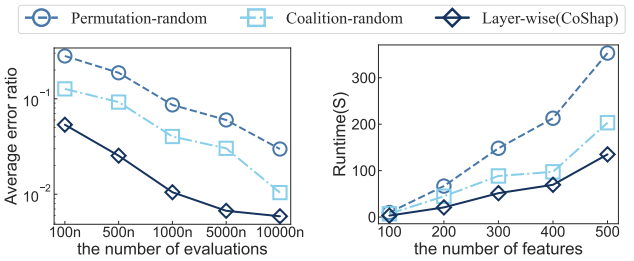
We conduct the ablation study to assess the effects of key components in CoShap, including the layer-wise coalition growth (layer-wise evaluation) and the feature-wise variance-aware allocation (feature-wise evaluation), on the Airport game dataset.

Layer-wise Evaluation. To explore the effects of layer-wise evaluation, we evaluate three different methods for calculating marginal contributions: 1) Permutation-based calculation using random



(a) Voting Game (b) Bank Marketing

Fig. 10. Effects of the m^* parameter on CoShap.

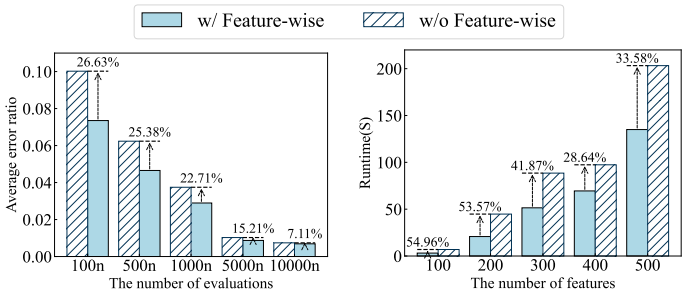


(a) Average error ratio (b) Runtime

Fig. 11. Effects of the layer-wise evaluation.

sampling (*Permutation-random*), 2) Coalition-based calculation using random sampling (*Coalition-random*), and 3) Layer-wise evaluation based on coalition growth (*Layer-wise*). In Figure 11, we observe that *Layer-wise* achieves the best performance in terms of err_{arg} and runtime. CoShap significantly reduces computational complexity by leveraging the structural properties of marginal contributions while maintaining accuracy comparable to both permutation-based and random sampling methods. Specifically, *Layer-wise* demonstrates superior computational efficiency over the permutation-based method, which becomes computationally intensive and time-consuming when processing large-scale datasets. Furthermore, compared to the random sampling method, *Layer-wise* achieves stability and reliability in performance by reusing results from upper-layer calculations. These findings highlight the substantial potential of CoShap for practical applications.

Feature-wise Evaluation. We assess the effects of feature-wise evaluation by comparing *w/ Feature-wise* (CoShap with the feature-wise variance-aware allocation) and *w/o Feature-wise* (CoShap without the feature-wise variance-aware allocation). Figure 12 (a) presents err_{ave} by varying m. The absence of feature-wise evaluation leads to a higher error for two main reasons. For one thing, the feature-wise evaluation focuses on top-k features with the highest variance, leading to more accurate estimates and reduced estimated error. For another, CoShap enables dynamic updates to evaluation allocation based on variance, optimizing budget distribution and improving the accuracy in the process. Figure 12 (b) presents the runtime of different numbers of features with $err_{ave} \leq 0.1$. By allocating samples based on the variance of marginal contributions, the optimized allocation method allows CoShap to converge to a specified error faster, saving evaluation budget and reducing runtime. This efficiency arises from focusing computational budgets on the top-k features with the highest variance, reducing the runtime required to achieve the desired accuracy. These results highlight the improvement of feature-wise evaluation in both faster convergence and lower computational costs.



(a) Average error ratio (b) Runtime

Fig. 12. Effects of the feature-wise evaluation.

6 Related Work

Shapley value approximation. The Shapley value computation is known to be an NP-hard problem since it involves evaluating the importance of features with all possible permutations [34, 47]. To address the challenge, several methods have been developed to approximate the Shapley value by evaluating a subset of randomly selected feature permutations, rather than the full factorial set. MC [5] proposes an unbiased, stochastic estimator by sampling permutations of features with probability $(1/n!)$ to compute the marginal contribution of each feature to the coalition's value. L-Shapley and C-Shapley [10] are two related approaches designed for structured data, such as images. Both approaches are biased Shapley value estimators as they restrict the game to consider only coalitions of features within the neighborhood of the feature being explained. IME [52] approximates the Shapley value with a fixed number of random permutations. A variant of IME, called adaptive sampling, improves convergence by allocating more samples to features with higher variance in their marginal contributions [53].

Explainable machine learning. In machine learning, the Shapley value is employed to quantify the contributions of input features to the output of a machine learning model at the instance level [8, 9]. Given a specific instance, the objective is to decompose the model's prediction and compute Shapley values for each feature of the instance. Shapley value methods for explainable machine learning could be broadly classified into two categories: domain-independent methods, which are applicable to any model, and domain-dependent methods, which are tailored to specific model types. Among domain-independent methods, the pioneering Shapley value-based universal explanation method SHAP [32] approximates the Shapley value with linear time. However, it operates under implicit assumptions about the features. To address the limitation, newer Shapley value-based explanation techniques [13, 16, 21, 22] have been proposed, with some methods generalizing SHAP beyond feature values to give attributions to first-order feature interactions [55, 56]. Domain-dependent methods, while less flexible than domain-independent approaches, offer significantly faster computation by leveraging the specific structures and internal properties of the model to enhance efficiency. These methods have been developed for various model types, including linear models, tree models, and deep learning models. First, LinearSHAP-based methods [32, 53] estimate Shapley value for linear models under the assumption that the data follows a multivariate Gaussian distribution. When this assumption does not hold, these methods may introduce bias. Second, TreeShap [31] is developed to interpret tree-based models such as decision trees, random forests, gradient-boosted trees, and their ensembles [36]. These models are capable of representing non-linear relationships and interaction effects, which makes them more complex than linear models. Finally, several early methods have been proposed to explain deep neural networks models, including

DeepLIFT [50], DASP [1] and ShapNets [58]. However, they may introduce bias in estimating Shapley values due to their reliance on backpropagation rules and local approximations [7].

7 Conclusions

In this paper, we identify the efficiency and reliability bottlenecks of existing Shapley value approximation methods. To address these issues, we introduce CoShap, a two-phase algorithm for model-agnostic interpretation. For efficiency, we leverage the pattern growth method to generate coalitions through layer-wise evaluation, effectively reducing redundant computations by merging identical marginal contributions and reusing coalition utilities. Recognizing the unreliable approximation of certain features, we propose a feature-wise evaluation that prioritizes top-k features with the highest variances, leading to more accurate and reliable estimations. We further present the variance-driven allocation method by dynamically distributing evaluation budgets, ensuring that CoShap maintains a balance between efficiency and accuracy. Our experiments demonstrate that CoShap delivers a 7.15× speedup in runtime and a 76.84% reduction in the average error ratio.

8 Acknowledgments

We thank the reviewers and the meta-reviewer for their valuable comments and suggestions, which help improve the quality of this paper. This work is partially supported by the NUS Faculty Development Fund. Jingxuan He's work is supported by the Tencent Rhino-Bird Elite Talent Foundation and the Bagui Young Top-notch Talent Fund. Xixian Han's work is supported by the National Natural Science Foundation of China (No. 62402135, U21A20513). Yanyan Shen's work is supported by the National Natural Science Foundation of China (No. 62522211).

References

[1] Marco Ancona, Cengiz Oztireli, and Markus Gross. 2019. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*. 272–281.

[2] Tian Bai, Shanshan Zhang, Brian L. Egleston, and Slobodan Vucetic. 2018. Interpretable Representation Learning for Healthcare via Capturing Disease Progression through Time. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 43–51.

[3] Shaofeng Cai, Kaiping Zheng, Gang Chen, HV Jagadish, Beng Chin Ooi, and Meihui Zhang. 2021. Arm-net: Adaptive relation modeling network for structured data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 207–220.

[4] Javier Castro, Daniel Gómez, Elisenda Molina, and Juan Tejada. 2017. Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation. *Computers & Operations Research* 82 (2017), 180–188.

[5] Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the Shapley value based on sampling. *Comput. Oper. Res.* 36, 5 (2009), 1726–1730.

[6] Centers for Disease Control and Prevention (CDC). 2023. Behavioral Risk Factor Surveillance System: 2022 BRFSS Survey Data and Documentation.

[7] Hugh Chen, Ian C. Covert, Scott M. Lundberg, and Su-In Lee. 2023. Algorithms to estimate Shapley value feature attributions. *Nat. Mac. Intell.* 5, 6 (2023), 590–601.

[8] Hugh Chen, Scott Lundberg, and Su-In Lee. 2021. Explaining models by propagating Shapley values of local components. *Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability* (2021), 261–270.

[9] Hugh Chen, Scott M Lundberg, and S Lee. 2021. Explaining a series of models by propagating local feature attributions. *CoRR abs/2105.00108* (2021).

[10] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2018. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610* (2018).

[11] Lingyang Chu, Xia Hu, Juhua Hu, Lanjun Wang, and Jian Pei. 2018. Exact and Consistent Interpretation for Piecewise Linear Neural Networks: A Closed Form Solution. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1244–1253.

[12] Xu Chu, John Morcos, Ihab F Ilyas, Mourad Ouzzani, Paolo Papotti, Nan Tang, and Yin Ye. 2015. Katara: A data cleaning system powered by knowledge bases and crowdsourcing. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1247–1261.

[13] Ian Covert and Su-In Lee. 2021. Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3457–3465.
[14] Lingxi Cui, Huan Li, Ke Chen, Lidan Shou, and Gang Chen. 2025. Nlctables: A dataset for marrying natural language conditions with table discovery. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3638–3647.
[15] Michele Dallachiesa, Amr Ebaid, Ahmed Eldawy, Ahmed Elmagarmid, Ihab F Ilyas, Mourad Ouzzani, and Nan Tang. 2013. NADEEF: a commodity data cleaning system. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 541–552.
[16] Susan Davidson, Daniel Deutch, Nave Frost, Benny Kimelfeld, Omer Koren, and Mikaël Monet. 2022. ShapGraph: An holistic view of explanations through provenance graphs and Shapley values. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 2373–2376.
[17] Yuhao Deng, Yu Wang, Lei Cao, Lianpeng Qiao, Yuping Wang, Jingzhe Xu, Yizhou Yan, and Samuel Madden. 2024. Outlier Summarization via Human Interpretable Rules. *Proceedings of the VLDB Endowment* 17, 7 (2024), 1591–1604.
[18] Daniel Deutch, Nave Frost, Amir Gilad, and Oren Sheffer. 2021. Explanations for data repair through shapley values. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 362–371.
[19] Anne M Euser, Carmine Zoccali, and Kitty J Jager. 2009. Cohort studies: prospective versus retrospective. *Nephron Clinical Practice* 113, 3 (2009), c214–c217.
[20] Ronald Aylmer Fisher. 1970. Statistical methods for research workers. In *Breakthroughs in statistics: Methodology and distribution*. Springer, 66–70.
[21] Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. 2020. Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272* (2020).
[22] Christopher Frye, Colin Rowat, and Ilya Feige. 2020. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems* 33 (2020), 1229–1239.
[23] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. *ACM SIGMOD International Conference on Management of Data* 29, 2 (2000), 1–12.
[24] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. 2019. Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1610–1623.
[25] Ahmet Kara, Dan Olteanu, and Dan Suciu. 2024. From Shapley Value to Model Counting and Back. *Proceedings of the ACM SIGMOD International Conference on Management of Data* 2, 2 (2024), 1–23.
[26] Patrick Kolpaczki, Viktor Bengs, Maximilian Muschalik, and Eyke Hüllermeier. 2024. Approximating the shapley value without marginal contributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 13246–13255.
[27] Weida Li and Yaoliang Yu. 2023. Faster Approximation of Probabilistic and Distributional Values via Least Squares. In *The 12th International Conference on Learning Representations*.
[28] Yifan Li, Xiaohui Yu, and Nick Koudas. 2024. Data Acquisition for Improving Model Confidence. *Proceedings of the ACM SIGMOD International Conference on Management of Data* 2, 3 (2024), 1–25.
[29] Stephen C Littlechild and GF Thompson. 1977. Aircraft landing fees: a game theory approach. *The Bell Journal of Economics* (1977), 186–204.
[30] Changshuo Liu, Lingze Zeng, Kaiping Zheng, Shaofeng Cai, Beng Chin Ooi, and James Wei Luen Yip. 2025. Neuralcohort: Cohort-aware neural representation learning for healthcare analytics. In *Forty-second International Conference on Machine Learning*.
[31] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence* 2, 1 (2020), 56–67.
[32] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 4765–4774.
[33] Xuan Luo, Jian Pei, Cheng Xu, Wenjie Zhang, and Jianliang Xu. 2024. Fast Shapley Value Computation in Data Assemblage Tasks as Cooperative Simple Games. *Proceedings of the ACM SIGMOD International Conference on Management of Data* 2, 1 (2024), 1–28.
[34] Sisi Ma and Roshan Tourani. 2020. Predictive and causal implications of using shapley value for model interpretation. In *Proceedings of the ACM SIGKDD workshop on causal discovery*. 23–38.
[35] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. 2013. Bounding the estimation error of sampling-based Shapley value approximation. *arXiv preprint arXiv:1306.4265* (2013).
[36] Masayoshi Mase, Art B Owen, and Benjamin Seiler. 2019. Explaining black box decisions by shapley cohort refinement. *arXiv preprint arXiv:1911.00467* (2019).
[37] Saumitra Mishra, Bob L Sturm, and Simon Dixon. 2017. Local interpretable model-agnostic explanations. In *ISMIR*, Vol. 53. 537–543.

[38] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31.
[39] Beng Chin Ooi, Shaofeng Cai, Gang Chen, Yanyan Shen, Kian-Lee Tan, Yuncheng Wu, Xiaokui Xiao, Naili Xing, Cong Yue, Lingze Zeng, et al. 2024. NeurDB: an AI-powered autonomous data system. *Science China Information Sciences* 67, 10 (2024), 200901.
[40] Guillermo Owen. 2013. *Game theory*. Emerald Group Publishing.
[41] Junyuan Pang, Jian Pei, Haocheng Xia, Xiang Li, and Jinfei Liu. 2025. Shapley Value Estimation based on Differential Matrix. *Proceedings of the ACM SIGMOD International Conference on Management of Data* 3, 1 (2025), 1–28.
[42] Chunjong Park, Anas Awadalla, Tadayoshi Kohno, and Shwetak N. Patel. 2021. Reliable and Trustworthy Machine Learning for Health Using Dataset Shift Detection. In *NeurIPS 2021*. 3043–3056.
[43] Kamil Pytlak. 2022. Indicators of Heart Disease. https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease.
[44] Amir Hossein Akhavan Rahnama, Laura Galera Alfaro, Zhendong Wang, and Maria Movin. 2024. Local Interpretable Model-Agnostic Explanations for Neural Ranking Models. *Swedish Artificial Intelligence Society* (2024), 151–159.
[45] Jacob Reiter. 2020. Developing an interpretable schizophrenia deep learning classifier on fMRI and sMRI using a patient-centered DeepSHAP. In *NeurIPS 2020*. 1–11.
[46] Theodoros Rekatsinas, Xu Chu, Ihab F Ilyas, and Christopher Ré. 2017. Holoclean: Holistic data repairs with probabilistic inference. *arXiv preprint arXiv:1702.00820* (2017).
[47] Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Oliver Kiss, Sebastian Nilsson, and Rik Sarkar. 2022. The Shapley Value in Machine Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022*, Luc De Raedt (Ed.). ijcai.org, 5572–5579.
[48] Henry Scheffe. 1999. *The analysis of variance*. John Wiley & Sons Publishing.
[49] P Schnohr, JS Jensen, H Scharling, and BG Nordestgaard. 2002. Coronary heart disease risk factors ranked by importance for the individual and community. A 21 year follow-up of 12000 men and women from The Copenhagen City Heart Study. *European heart journal* 23, 8 (2002), 620–626.
[50] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*. 3145–3153.
[51] Michelle Si and Jian Pei. 2024. Counterfactual Explanation of Shapley Value in Data Coalitions. *Proceedings of the VLDB Endowment* 17, 11 (2024), 3332–3345.
[52] Erik Strumbelj and Igor Kononenko. 2010. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research* 11 (2010), 1–18.
[53] Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems* 41 (2014), 647–665.
[54] Qiheng Sun, Jiayao Zhang, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2024. Shapley value approximation based on complementary contribution. *IEEE Transactions on Knowledge and Data Engineering* (2024).
[55] Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. 2020. The shapley taylor interaction index. In *International conference on machine learning*. 9259–9268.
[56] Mukund Sundararajan and Amir Najmi. 2020. The many Shapley values for model explanation. In *International conference on machine learning*. 9269–9278.
[57] Fei Tian, Lan Chen, Zhengmin Min Qian, Hui Xia, Zilong Zhang, Jingyi Zhang, Chongjian Wang, Michael G Vaughn, Maya Tabet, and Hualiang Lin. 2023. Ranking age-specific modifiable risk factors for cardiovascular disease and mortality: evidence from a population-based longitudinal study. *EClinicalMedicine* 64 (2023).
[58] Rui Wang, Xiaoqian Wang, and David I Inouye. 2021. Shapley explanation networks. *arXiv preprint arXiv:2104.02297* (2021).
[59] Tingting Wang, Shixun Huang, Zhifeng Bao, J. Shane Culpepper, Volkan Dedeoglu, and Reza Arablouei. 2024. Optimizing Data Acquisition to Enhance Machine Learning Performance. *Proceedings of the VLDB Endowment* 17, 6 (2024), 1310–1323.
[60] Yatong Wang, Yuncheng Wu, Xincheng Chen, Gang Feng, and Beng Chin Ooi. 2023. Incentive-aware decentralized data collaboration. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–27.
[61] Eyal Winter. 2002. The shapley value. *Handbook of game theory with economic applications* 3 (2002), 2025–2054.
[62] Yuncheng Wu, Naili Xing, Gang Chen, Tien Tuan Anh Dinh, Zhaojing Luo, Beng Chin Ooi, Xiaokui Xiao, and Meihui Zhang. 2023. Falcon: A privacy-preserving and interpretable vertical federated learning system. *Proceedings of the VLDB Endowment* 16, 10 (2023), 2471–2484.
[63] Xinyi Xu, Thanh Lam, Chuan Sheng Foo, and Bryan Kian Hsiang Low. 2024. Model shapley: equitable model valuation with black-box access. *Advances in Neural Information Processing Systems* 36 (2024).
[64] Jiayao Zhang, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, et al. 2023. Efficient Sampling Approaches to Shapley Value Approximation. *Proceedings of the ACM SIGMOD International Conference on Management of Data* 1, 1 (2023),

48:1–48:24.

[65] Xianli Zhang, Buyue Qian, Shilei Cao, Yang Li, Hang Chen, Yefeng Zheng, and Ian Davidson. 2020. INPREM: An Interpretable and Trustworthy Predictive Model for Healthcare. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 450–460.

[66] Zhanhao Zhao, Shaofeng Cai, Haotian Gao, Beng Chin Ooi, and Others. 2025. NeurDB: On the Design and Implementation of an AI-powered Autonomous Database. In *Proceedings of the 15th Conference on Innovative Data Systems Research (CIDR '25)*. www.cidrdb.org, Amsterdam, Netherlands.

[67] Kaiping Zheng, Shaofeng Cai, Horng Ruey Chua, Wei Wang, Kee Yuan Ngiam, and Beng Chin Ooi. 2020. Tracer: A framework for facilitating accurate and interpretable analytics for high stakes applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 1747–1763.

Received October 2025; revised January 2026; accepted February 2026