Unlocking the Game: Estimating Games in Möbius Representation for Explanation and High-Order Interaction Detection

Majid Mohammadi, Ilaria Tiddi, Annette Ten Teije

Department of Computer Science, Vrije Universiteit Amsterdam
De Boelelaan 1111, 1081 HV Amsterdam
majid.mohammadi690@gmail.com, i.tiddi@vu.nl, annette.ten.teije@vu.nl

Abstract

Shapley value-based explanations are widely utilized to demystify predictions made by opaque models. Approaches to estimating Shapley values often approximate explanation games as inessential and estimate the Shapley value directly as feature attribution with a limited capacity to quantify feature interactions. This paper introduces a new approach for calculating Shapley values that relaxes the assumption of inessential games and is proven to provide additive feature attribution. The initial formulation of the proposed approach includes the estimation of game values in their Möbius representation with exponentially many parameters, but we put forward a polynomial-time algorithm designed to manage the game's numerous values and achieve an efficient linear-time computation of the Shapley value. Moreover, this formulation uniquely enables identifying only the significant high-order feature interactions amidst a potentially exponential set. Through experiments, we demonstrate the robust performance of our methodology in game estimation and in providing explanations for multiple black-box models.

1 Introduction

Feature attributions serve as a key technique for interpreting opaque machine learning models, offering insights into the inner mechanism of these complex systems. Shapley value-based explanation is a popular technique for feature attribution (Lundberg and Lee 2017; Covert, Lundberg, and Lee 2020; Fumagalli et al. 2024; Štrumbelj and Kononenko 2014). A primary obstacle in determining the Shapley value arises from its inherent computational complexity, which grows exponentially with the addition of features. In an effort to manage this complexity, different strategies such as Monte Carlo estimation and linear regression, are employed to approximate the Shapley values by sampling from the characteristic function (see (Chen et al. 2023) for a survey).

Despite their widespread use in various domains, the current methods estimate an inessential game for explanation (Kumar et al. 2021; Covert and Lee 2021), and their ability to quantify the interactions among features remains limited (Lundberg and Lee 2017; Tsai, Yeh, and Ravikumar 2023; Masoomi et al. 2021). This implies that the applications of current methods are largely confined to solving the problem of attribution - that is, assessing the significance of singular features. However, it falls short of illuminating the dynamics of how these features interact, which is critical for understanding the model's synergistic effects.

In this study, we introduce a novel regression problem aimed at estimating the Möbius representation of the characteristic function to explain the prediction of a black-box model for individual instances. The proposed model is proven to provide an additive feature attribution while approximating the explanation game as an essential game, thereby relaxing the inessential game presumption for the Shapley value estimation. In addition, the Möbius representation allows for the identification of high-order interactions between features. Given that both the characteristic function and its Möbius representation encompass a large number of parameters, tackling this regression task poses significant computational hurdles. Nevertheless, we have developed a polynomial-time algorithm that remarkably reduces the computational burden from $O(2^d)$ parameters to $O(n)$, where d and n are the number of features and samples, respectively. Additionally, our approach enables the detection of significant feature interactions of any order directly from the regression's outcome, obviating the need to enumerate potentially exponential interactions. Furthermore, we explore optimal regularization formulation to provide parsimonious models for explanation and feature interaction.

We therefore present GEM-FIX (**G**ame **E**stimation in **M**öbius representation for **F**eature **I**nteraction detection and e**X**planation), with the principal contributions being enumerated as follows: (i) We put forward a framework for estimating efficiently the Möbius representation of games given a subset of characteristic values; (ii) We develop a new representation of the Shapley value, enabling the computation of feature Shapley values in linear time from the solution of the proposed regression problem; (iii) We develop an efficient method for identifying only the significant (non-zero) interactions from among potentially exponentially many, through the solution to our regression model. This approach prompts us to suggest specific regularization strategies for both interactions and Shapley values. The GEM-FIX implementation is publicly available at https://github.com/Majeed7/GEMFIX.

2 Background

Notation In this paper, we denote the set of d features as D, and represent its power set, which encompasses all non-empty subsets of D, by $\mathfrak{D}$. To facilitate the computation of Shapley values, we select a subset from $\mathfrak{D}$, referred to as $\mathfrak{S}$. We assume that $\{\emptyset, D\} \notin \mathfrak{S}$, and define $\mathfrak{S}^+ = \mathfrak{S} \cup \{D\}$. Each sample S within this subset $\mathfrak{S}$ is encoded using a binary vector $\mathbf{z}_S \in \mathbb{R}^{|\mathfrak{D}|}$. The components of this vector are denoted as $z_{SS'}$ for each $S' \in \mathfrak{D}$, with the convention that $z_{SS'} = 1$ only when S' is a subset of S. In instances where vectors or matrices are indexed by subsets such as S, this indexing scheme is indicative of the elements corresponding to subsets of features within $\mathfrak{D}$ or $\mathfrak{S}$. Consequently, a vector $\mathbf{z}_S$ is conceptualized as a collection of elements, explicitly defined as $\left(z_{SS'}\right)_{S' \in \mathfrak{D}}$. Also, the notation $|.|$ signifies set cardinality, and $\|\cdot\|_p$ is employed to represent the p-norm. The vectors are shown by bold-faced lower-case letters, the sets with capital letters, and matrices with capital Greek letters.

Shapley Value and Möbius Representation A cooperative game is identified by a characteristic function $v : 2^D \to \mathbb{R}$ that assigns a value for each subset $S \subseteq D$. A game is said to be *inessential* if $v(S) = \sum_{i \in S} v(\{i\}), \forall S \in \mathfrak{D}$ (Kumar et al. 2021). Given a characteristic function, there are different solution concepts in game theory that attempt to provide a fair distribution of the payoff of the grand coalition (i.e., $v(D)$) among the players involved. The Shapley value is an axiomatic approach that is guaranteed to satisfy four important axioms: efficiency, null player, symmetry, and linearity (Shapley 1953). For a given characteristic function v, the Shapley value for player i, denoted by $\phi_i(v)$, is defined as (Shapley 1953):

$$\phi_i(v) = \sum_{S \subseteq D \setminus \{i\}} \frac{|S|!(|D| - |S| - 1)!}{|D|!} \Big[v\left(S \cup \{i\}\right) - v(S) \Big]$$

The term $v\left(S \cup \{i\}\right) - v(S)$ represents the marginal contribution of player i to a coalition S. This quantifies the added value player i brings to the coalition, which is pivotal in calculating the Shapley value. The fraction preceding this term normalizes the contribution across all permutations of players joining the coalition, ensuring every potential sequence of players joining is equally considered.

A useful representation of the Shapley value is computed by using the Möbius representation. The set function v can be represented by the Möbius representation m as (Grabisch 1996):

$$v(B) = \sum_{A \subseteq B} m(A), \quad \forall B \subseteq D, \tag{1}$$

and the Möbius representation m can be written as:

$$m(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} v(B). \tag{2}$$

The Möbius representation provides a powerful tool for decomposing the characteristic function into a sum of contributions from different coalitions. Defining $m(A)$ as above effectively captures the interaction among subsets of players (or subsets of features), allowing for a more granular analysis of their contributions. This formulation is especially useful in efficiently computing the Shapley value by directly summing over subsets containing the player of interest. In particular, the Shapley value for feature i can be represented by using the Möbius representation as (Grabisch 1996):

$$\phi_i = \sum_{B \subseteq D | i \in B} \frac{1}{|B|} m(B). \tag{3}$$

With a tolerable abuse of notation, we henceforth refer to $v(A)$ as $m(A)$ as v_A and m_A, respectively.

Kernel SHAP Many methods utilize cooperative game theory and the Shapley value for interpreting black-box machine learning models. These methods sample subsets of features and estimate the characteristic function for each subset. Specifically, for the local explanation of a sample $\boldsymbol{x}$, Kernel SHAP estimates the characteristic function for a feature subset S as follows (Lundberg and Lee 2017):

$$v(S) = \mathbb{E}\left[f(\boldsymbol{x}_S, X_{D \setminus S}) \right], \tag{4}$$

where $f(\boldsymbol{x}_S, X_{D \setminus S})$ represents the model's prediction, evaluated with features in subset S set to their values in $\boldsymbol{x}$ and the remaining features $D \setminus S$ replaced with data from the reference dataset X. We refer to equation (4) as the *explanation game*. The essence of Kernel SHAP lies in its approach to decomposing the model prediction into contributions from each feature, reflecting the notion of *additive feature attribution*. This approach implies that the prediction for a particular instance can be expressed as a sum of the effects of each feature on the prediction, formalized as:

$$f(\boldsymbol{x}) = \phi_0 + \sum_i \phi_i, \tag{5}$$

where ϕ_i are the Shapley values representing the contribution of feature i to the prediction, and ϕ_0 represents the prediction value when no features are present, effectively acting as the average prediction over the dataset. To estimate the Shapley values, Kernel SHAP employs a weighted linear regression (Lundberg and Lee 2017):

$$\min_{\phi_1, \ldots, \phi_d} \sum_{S \in \mathfrak{S}} w(S) \left(\phi_0 + \sum_{i \in S} \phi_i - v_S \right)^2, \tag{6}$$

subject to the constraint that $\phi_0 + \sum \phi_i = v_D$, with $w(S)$ defined as:

$$w(S) = \frac{d - 1}{\binom{d}{|S|} |S| (d - |S|)}, \tag{7}$$

and setting $w(\{\emptyset\}) = w(D) = \infty$. This formulation, including the constraint, ensures the model's prediction is decomposed into a base value (ϕ_0) and feature contributions (ϕ_i). In addition, defining a game $\hat{v}$ by $\hat{v}(S) = \phi_0 + \sum_{i \in S} \phi_i$, it indicates that Kernel SHAP effectively constructs an *inessential game* for the purpose of providing explanations (Covert and Lee 2021; Kumar et al. 2021).

3 Unlocking the Game: Estimating Möbius Representation

Linear Regression for Möbius Representation Estimation As discussed, Kernel SHAP and similar methods for explainability focus on explaining the prediction of an instance by an inessential approximation of the explanation game and directly computing the Shapley value. We now put forward an approach that estimates the characteristic function in their Möbius representation, where it provides an additive explanation by approximating explanation games as essential.

For each $S \in \mathfrak{S}$, the value v_S represents the prediction of the model when only the features in S are considered as shown in equation (4). The Möbius representation of v_S is:

$$v_S = m_\emptyset + \sum_{S' \subseteq S, S' \neq \emptyset} m_{S'}, \tag{8}$$

where $m_\emptyset$ and $m_{S'}$ are the Möbius representation of null game and coalition S', respectively. Based on equation (8), we now put forward a new regression to find the values of m_0 and $m_{S'}$ based on v_S:

$$\begin{aligned} &\min_{\{m_S\}_{S \in \mathfrak{D}}} \sum_{S \in \mathfrak{S}} \big(m_\emptyset + \sum_{S' \subseteq S, S' \neq \emptyset} m_{S'} - v_S\big)^2 \\ &s.t. \quad m_\emptyset + \sum_{S \in \mathfrak{D}} m_S = v_D, \end{aligned} \tag{9}$$

where $m_\emptyset = v_\emptyset$. Solving this regression problem provides the Möbius representation of all the game values v, with which one can compute the Shapley value as well as potential interactions. Before discussing how to compute such values efficiently, we first discuss two important properties of this approach. These two properties are straightforward to verify based on minimization (8), but proofs are also provided for further clarification. All the proofs are placed in Appendix A.

Property 1 *The local explanation for $\boldsymbol{x}$ provided by model* (9) *is an additive feature attribution as in equation* (5)*.*

Property 2 *The explanation game estimated by equation* (8) *is essential.*

Regularized Formulation and Efficient Solution The major challenge with minimization (9) arises from the exponentially many parameters involved, complicating the task of finding an efficient solution. To address this issue, we implement two modifications to the optimization framework. We introduce L2 regularization on the Möbius representation to manage its complexity, and augment the objective function with a penalty for any deviation from the equality constraint for simplicity. Thus, it follows:

$$\min_{\boldsymbol{m}, \boldsymbol{e}} \frac{\lambda}{2} \sum_{S \in \mathfrak{S}^+} \xi_S e_S^2 + \frac{1}{2} \|\boldsymbol{m}\|_2^2 \quad \text{s.t. } e_S = \boldsymbol{m}^T \boldsymbol{z}_S + m_\emptyset - v_S \tag{10}$$

where $\xi_S = 1$ except for $\xi_D = \infty$, $\boldsymbol{m} = \big(m_S\big)_{S \in \mathfrak{D}}$, $\boldsymbol{z}_S = \big(z_{ST}\big)_{T \in \mathfrak{D}}$ is a binary vector. First, we present the solution to this problem in the following theorem.

Theorem 1 *The solution to minimization* (10) *is $\boldsymbol{m} = \sum_{S \in \mathfrak{S}^+} \alpha_S \boldsymbol{z}_S$, where $\boldsymbol{\alpha} \in R^{|\mathfrak{S}^+|}$ is the solution to the following system of linear equation ($\forall S, T \in \mathfrak{S}^+$):*

$$\Omega \boldsymbol{\alpha} = \boldsymbol{v}, \quad s.t.\ \Omega_{ST} = \boldsymbol{z}_S^T \boldsymbol{z}_T + \boldsymbol{I}_{S=T}(1/\lambda \xi_S), \tag{11}$$

where $\Omega \in R^{|\mathfrak{S}^+| \times |\mathfrak{S}^+|}$, $\boldsymbol{v} = \big(v_S - m_\emptyset\big)_{S \in \mathfrak{S}^+}$, and $\boldsymbol{I}_{S=T}$ is the indicator function and is one if and only if $S = T$.

Since Ω is evidently positive definite, the solution to the system of linear equations in Theorem 1 can be efficiently computed by using Cholesky decomposition (Strang 2012). The computation of Ω is, however, of exponential complexity, stemming from the inner product $\boldsymbol{z}_S^T \boldsymbol{z}_{S'}$ where $\boldsymbol{z}_S, \boldsymbol{z}_{S'} \in R^{|\mathfrak{D}|}$. We now show that such an inner product can be computed efficiently in $O(d)$.

Theorem 2 *For two subsets $S, S' \in \mathfrak{S}^+$ with their binary encoded vectors $\boldsymbol{z}_S$ and $\boldsymbol{z}_{S'}$, we have*

$$\boldsymbol{z}_S^T \boldsymbol{z}_{S'} = 2^{|S \cap S'|} - 1 \tag{12}$$

4 Shapley Value and Interaction Detection Estimation

This section efficiently calculates the Shapley value and identifies significant interactions in an efficient manner. We also put forward several regularization techniques that are useful for explanation.

4.1 Shapley Value Estimation

The solution to problem (10) is efficient, but we require to obtain $\boldsymbol{m}$ in order to compute the Shapley value and significant interactions. Due to the exponential number of elements in $\boldsymbol{m}$, directly calculating Shapley values or detecting interactions through $\boldsymbol{m}$ is impractical, especially for large feature sizes. To overcome this, we introduce a theorem that serves as the basis for computing Shapley values and significant interactions directly from $\boldsymbol{\alpha}$, thereby circumventing the calculation of $\boldsymbol{m}$.

Theorem 3 *Let $\boldsymbol{\alpha}$ be the solution to minimization* (10)*, then $m_{S'} = \sum_{\mathfrak{S}^+ \ni S \supseteq S'} \alpha_S, \forall S' \in \mathfrak{D}$.*

We now present a new representation for the Shapley value based on $\boldsymbol{\alpha}$.

Theorem 4 *Let $\boldsymbol{\alpha}$ be the optimal to problem* (10)*, then ϕ_i is computed as:*

$$\phi_i = \sum_{S \in \mathfrak{S}^+ | i \in S} \Big(\sum_{\theta=1}^{|S|} \frac{1}{\theta} \binom{|S|-1}{\theta-1}\Big) \alpha_S \tag{13}$$

This theorem indicates that the Shapley value can be computed by using only $|\mathfrak{S}^+|$ parameters, in contrast to the crude computations with the prerequisite of 2^d parameters. Also, the time complexity of equation (13) is linear in $O(|\mathfrak{S}^+|)$.

4.2 Interaction Detection Framework

One advantage of the formulation in problem (9) allows detecting feature interactions by finding significant m_S's. However, the challenge of identifying significant interactions persists due to the exponentially many potential interactions. To tackle this, we propose a methodology that simplifies the process of detecting high-order interactions by introducing a regularization scheme over $\boldsymbol{\alpha}$.

Proposition 1 *For any subset S within $\mathfrak{D}$, if $\alpha_{S'} = 0$ for all $S' \supseteq S$, then no interactions exist for the feature set S.*

Using this proposition, along with Theorem 3, all the significant interactions can be computed from the non-zero elements in $\boldsymbol{\alpha}$, instead of iterating over all the potential interactions. We can start with an empty set of interactions and add significant interactions $S' \subseteq S$ when α_S is significant (see Algorithm 1). In addition, this proposition along with Theorem 4 suggests that a sparse $\boldsymbol{\alpha}$ would also minimize the number of features with a significant Shapley value. We therefore propose the following optimization problem to find a sparse solution $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\Omega\boldsymbol{\alpha} - \boldsymbol{v}\|_2^2 + \lambda'\|\Gamma\boldsymbol{\alpha}\|_1, \tag{14}$$

where $\lambda' > 0$ is a regularization parameter, and Γ is a matrix that induces sparsity in the significant interactions and influential features. We now discuss multiple choices for Γ.

Sparsity and Interaction Regularization Proposition 1 outlines a way for deducing feature interactions from $\boldsymbol{\alpha}$; computing m_S for every S within $\mathfrak{D}$ remains prohibitively expensive due to the number of all interactions, despite many of m_S being null for a sparse $\boldsymbol{\alpha}$. The proposition highlights that only the non-zero elements of $\boldsymbol{\alpha}$ correspond to significant interactions, indicating that a sparser $\boldsymbol{\alpha}$ will lead to fewer significant interactions. Consequently, employing $\Gamma = I$ (where I represents the identity matrix) yields a sparse $\boldsymbol{\alpha}$ and, by extension, fewer significant interactions.

This approach refines the process of detecting significant interactions from a cumbersome exhaustive search of all possible interactions to a focused examination of non-zero elements in $\boldsymbol{\alpha}$. For instances where $\boldsymbol{\alpha}$ contains numerous non-zero elements, a greedy algorithm can be utilized. Such an algorithm prioritizes $\boldsymbol{\alpha}$ values by their magnitude, highlighting the most significant interactions first, and terminates based on a predefined time constraint, streamlining the identification of key interactions.

Sparsity and Shapley Regularization In certain scenarios, the explanations are clearer with minimal significant features (Ribeiro, Singh, and Guestrin 2016). This objective can be accomplished by introducing penalties over the Shapley values within the optimization framework, such as applying regularization to $\boldsymbol{\phi}$ in the Kernel SHAP minimization (6). We now show that it is possible to attain sparse Shapley values by implementing regularization on $\boldsymbol{\alpha}$.

Lemma 1 *Problem* (14) *leads to sparse Shapley values with* Γ *be defined as*

$$\Gamma_{iS} = \begin{cases} 0, & \text{if } i \notin S, \\ \sum_{\theta=1}^{|S|} \frac{1}{\theta}\binom{|S|-1}{\theta-1}, & \text{otherwise}, \end{cases} \qquad \Gamma \in \mathbb{R}^{d \times |\mathfrak{S}^+|} \tag{15}$$

4.3 Overall Algorithm and Complexity

Algorithm 1 summarizes the overall procedure for explaining model f for a given instance $\boldsymbol{x}$. The procedure includes generating set $\mathfrak{S}^+$ and their value $v_S, \forall S \in \mathfrak{S}^+$, for which we used the Kernel SHAP sampling procedure. We then need to compute Ω, obtaining $\boldsymbol{\alpha}$, estimating the Shapley values $\boldsymbol{\phi}$, and the significant interactions with a greedy approach.

Algorithm 1 GEM-FIX: **G**ame **E**stimation in **M**öbius Representation for **F**eature **I**nteraction and e**X**planation

Require A trained model f over d features, an instance $\boldsymbol{x}$
Generate set $\mathfrak{S}^+ \subseteq \mathfrak{D}$
Compute v_S according to equation (4) for $S \in \mathfrak{S}^+$
Compute matrix Ω using Theorem 2
Compute $\boldsymbol{\alpha}$ as in Theorem 3 or in minimization (14)
$\boldsymbol{\phi} = zeros(d)$ {Shapley value initialization with zeros}
for $i = 1, ..., d$ **do**
  $\phi_i = \sum_{S \in \mathfrak{S}^+ | i \in S} \left(\sum_{\theta=1}^{|S|} \frac{1}{\theta}\binom{|S|-1}{\theta-1} \right) \alpha_S$
end for
interactions = { } {dictionary for significant interactions}
$\hat{\boldsymbol{\alpha}} = sort(|\boldsymbol{\alpha}|)$ {Sort $\boldsymbol{\alpha}$ by descending magnitude}
while $|\hat{\alpha}_S| > \epsilon$ and !timeover() **do**
  for $S \subseteq S$ **do**
    interactions[S'] += $\hat{\alpha}_S$
  end for
end while
Return $\boldsymbol{\phi}$, interactions

The algorithm's overall time complexity involves several computations. Initially, we compute the matrix Ω, with a complexity of $O(|\mathfrak{S}|^{+^2} d)$, assuming intersection computation between sets at worst-case $O(d)$. The complexity of deriving $\boldsymbol{\alpha}$ using Cholesky decomposition is $O(|\mathfrak{S}|^{+^3})$ at its peak, and computing the Shapley values for all d features requires $O(|\mathfrak{S}|^{+})$. Additionally, significant interactions can be computed based on non-zero elements of $\boldsymbol{\alpha}$ and the corresponding set size of non-zero $\boldsymbol{\alpha}$. Let $n_{\boldsymbol{\alpha}}$ represent the number of non-zero elements in $\boldsymbol{\alpha}$, and s' denote the average set size among non-zero elements in $\boldsymbol{\alpha}$. Consequently, all interactions can be computed in $O(n_{\boldsymbol{\alpha}} 2^{s'})$. Thus, significant interactions can be efficiently computed for sparse $\boldsymbol{\alpha}$ or small s'.

5 Related Works

Aside from Kernel SHAP, the Shapley value has been used for local explanation (Lundberg and Lee 2017; Covert and Lee 2021; Mohammadi, Tiddi, and Ten Teije 2023; Breuer et al. 2024). QII (Datta, Sen, and Zick 2016) is a method for local explanation that uses game-theoretic notions (Shapley and Banzhaf values) for explanation. Methods like *Bivariate-SHAP* (Masoomi et al. 2021) focus on unraveling pairwise feature interactions, offering a more targeted analysis of how two features influence predictions in tandem. In addition, approaches such as L2X (Chen et al. 2018a) and C-Shapley (Chen et al. 2018b) harness mutual information to detect and quantify the strength of feature interactions. The issue with such approaches is that interactions are limited up to a cer-

tain order, in contrast to GEM-FIX, which is able to detect high-order interactions.

There are also different interaction indices that explain the interactions among features. One of the first works is the Shapley interaction index (Grabisch and Roubens 1999), where the notion of the Shapley value on averaging the marginal contributions of a feature is extended to the marginal contribution of a feature subset. Other indices such as Shapley-Taylor (Sundararajan, Dhamdhere, and Agarwal 2020) and Faith-SHAP (Tsai, Yeh, and Ravikumar 2023) are introduced as well, grounded in different axioms. GEM-FIX can identify the most important interacting features through the Möbius representation among an exponentially big set, but such interaction indices could also be used along with GEM-FIX to measure the level of interactions for the identified set from GEM-FIX (the choice is up to the use case and the suitability of the interaction indices).

6 Experiments

This section compares GEM-FIX with several methods: Kernel SHAP (Lundberg and Lee 2017), Unbiased SHAP (Covert and Lee 2021), Sampling SHAP (Štrumbelj and Kononenko 2014), Bivariate SHAP (Masoomi et al. 2021), LIME (Ribeiro, Singh, and Guestrin 2016), and MAPLE (Plumb, Molitor, and Talwalkar 2018). Due to the space limit, we place some details on the experiments in the appendix B.

6.1 Game Simulation

For the first experiment, we evaluated the performance of GEM-FIX in estimating the Shapley value from a subset of characteristic functions in a cooperative game. To that end, we simulate cooperative games by randomly generating the characteristic function and computing the exact Shapley value accordingly (the procedure for generating games is elaborated on in Appendix B.1). We then draw samples from the generated characteristic function and subject them to GEM-FIX and SHAP to estimate the Shapley value. Specifically, we explore two versions of our proposed approach: one employing a linear model as in minimization (9) (*GEM-FIX-LR*), and another incorporating regularization into the minimization process as in equation (10). Following that, we compare our methods with the standard Kernel SHAP, as well as a variation of Kernel SHAP with L2 regularization, denoted as *SHAP-L2*, to verify the regularization effect on the solution. We set $\lambda = 10^3$ for both GEM-FIX and SHAP-L2 to reduce the potential regularization effect on the estimation.

Figure 1 illustrates the absolute difference between the actual Shapley values and those estimated by methods against the number of samples drawn for the essential games. Both variants of GEM-FIX demonstrate a superior capability to more accurately estimate the Shapley values compared to Kernel SHAP (though GEM-FIX-LR cannot generate the estiamtion if the player count exceeds 13, see Appendix B.1 for more experiments). This discrepancy is more pronounced with smaller sample sizes, where SHAP struggles to yield reliable estimates of the Shapley value. Further details on experiments with several other players counts and comparison on inessential games are available in Appendix B.1.

6.2 Synthesized Experiments

We evaluate different explainable methods on three synthesized datasets with known important features (see Appendix B.2 for the dataset generation procedure). For each dataset, we generate 100 samples with ten features and subject all the samples to explainable methods to identify the most influential features. We then contrast the identified features by explainable methods to those of the ground truth and compute the average rank of the influential features for each sample. The methods are then compared based on the mean and the standard deviation of the average rank across different samples. The more effective the explainable method, the lower the mean of average ranks.

Figure 2 shows the box plots of the mean rank of explainable methods for the three synthesized experiments. For the first datasets (ideally, the average rank is 2, see Appendix B.2), the Shapley value-based explanations show superior performance, while in the last two (the ideal average rank is 2.5 and 3 for the second and third dataset, respectively - see Appendix B.2), GEM-FIX is particularly competitive or superior to all the methods. In addition, GEM-FIX demonstrates good performance regarding the standard deviation of average ranks, implying that it has recovered the most influential features more consistently across different samples.

We also conducted an evaluation of GEM-FIX's ability to detect feature interactions using a synthetic experiment. We generated 120 datasets with binary features designed to mimic random subset generation in explanations. Each dataset's response variable was set as the weighted sum of individual feature values plus six randomly created interaction terms. Specifically, interaction terms were generated for the second, third, and fourth orders. For each interaction order, two distinct terms were created. We applied GEM-FIX to these datasets and identify all the significant interactions above a threshold of 0.001, and compared the detected interactions with ground truth. Table 1 presents the true positive (TP), false positive (FP), and false negative (FN) counts for the interaction terms detected across various interaction orders. In particular, GEM-FIX demonstrated high TP rates, successfully identifying many interaction terms. The FP rates were also very acceptable, but slightly higher when the data set contained overlapping interactions, such as the terms {1,2,3} and {1,2}. In such cases, GEM-FIX identified additional interactions like {2,3} and {1,3}. Although these interactions were technically incorrect, they were not entirely inaccurate as there is an interaction among the detected terms. This overlap primarily contributed to the higher FP observed in second-order interactions and the high FN rates in fourth-order interactions. Despite these issues, GEM-FIX performance is promising in identifying interactions, indicated by the median statistics showing promising recovery rates for the most significant interactions, especially in the second and third orders.

6.3 Tabular Datasets

In our real experimental setup, we utilize several real-world tabular datasets (detailed information provided in Appendix B.3). For each dataset, a random forest model comprising

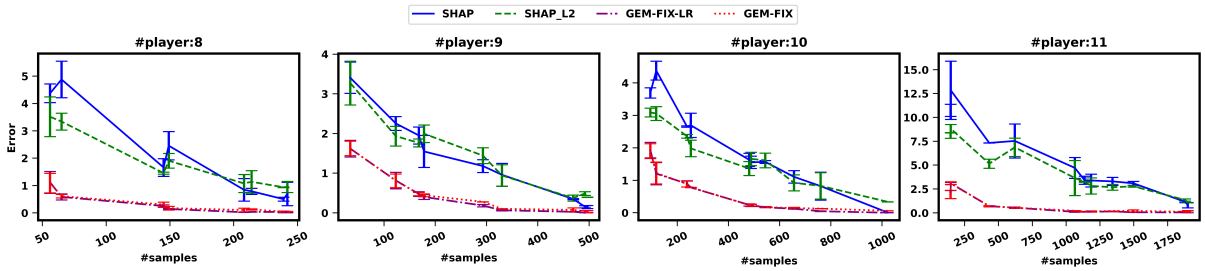

Figure 1: The absolute deviation from ground truth Shapley value for the game simulation.

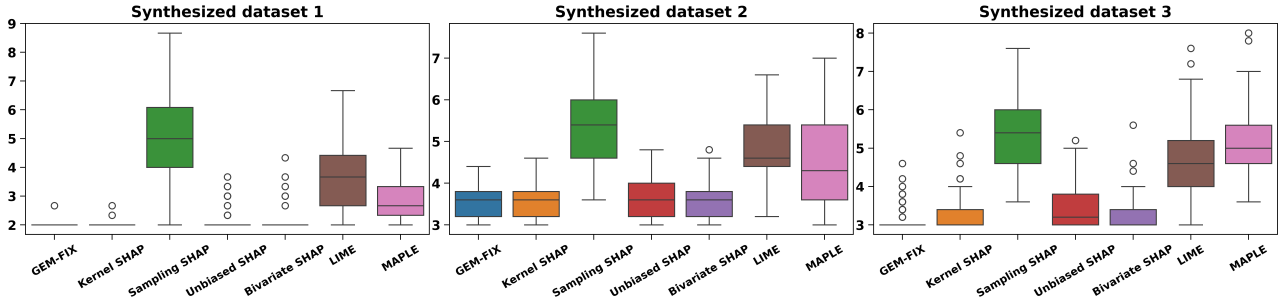

Figure 2: The box plot of the most influential features for the three synthesized datasets.

Interaction Order (#Total)	Statistic	TP	FP	FN
Second (240)	Sum	236	43	4
	Median	2	0	0
	Mean	1.96	0.09	0.033
Third (240)	Sum	228	32	12
	Median	2	0	0
	Mean	1.9	0.26	0.1
Fourth (240)	Sum	160	19	80
	Median	1	0	1
	Mean	1.33	0.15	0.66

Table 1: Results of the synthetic experiment for interaction detection across different interaction orders, summarizing the sum, median, and mean of TP, FP, and FN from 120 experiments. Each experiment included six interaction terms of orders two, three, and four.

500 trees is trained. Subsequently, this model and a set of 50 randomly selected samples undergo analysis using various explainable methods to determine key features for the samples. Also, we calculate the exact Shapley values for each sample to establish a ground truth and compare different methods based on their deviations from these values. This experiment incorporates Shapley value-based explanations only due to the comparative analysis focusing on the deviation from the ground truth Shapley value.

Figure 3 shows the mean and standard deviation of discrepancies to the exact Shapley values across different sample sizes. According to the graph, GEM-FIX exhibits superior accuracy in terms of average deviation from the ground-truth Shapley values, particularly with smaller sample sizes. Conversely, other methods perform much better with increasing sample sizes, as indicated by a more significant reduction in deviation from the ground truth as the sample size increases. The time comparison of the explainable methods on tabular datasets is also provided in Appendix B.3.

6.4 Textual Datasets

In this section, we aim to explore the effect of masking words identified as least important by various explainable methods on the predictions of a pre-trained BERT model used for sentiment analysis. We analyze three textual datasets, applying explainable methods to 50 reviews from each dataset. Further details about the BERT model and the datasets are provided in Appendix B.4. To evaluate different explainability methods, we identify influential words in each review, masking those deemed least impactful. We hypothesize that masking these less influential words will not significantly alter the model's predictions. Variations in predictions, with varying percentages of words masked, are illustrated in Figure 4, where the x-axis represents the percentage of words masked. The results show that GEM-FIX performs superiorly by accurately identifying both the most and the least influential words, as evidenced by the stable predictions when significant portions of reviews are masked. Kernel SHAP and Sampling SHAP also demonstrate competitive performance, whereas LIME and MAPLE appear less effective in pinpointing critical words in reviews. Additionally, we assess these methods based on the time efficiency of generating explanations. According to Figure 5, GEM-FIX's execution time is on par with Kernel SHAP and outperforms Sampling SHAP, while, in principle, it solves a problem with exponentially many parameters. In contrast, LIME and MAPLE, despite their speed, offer less reliable results as indicated by their performance in the word removal analysis.

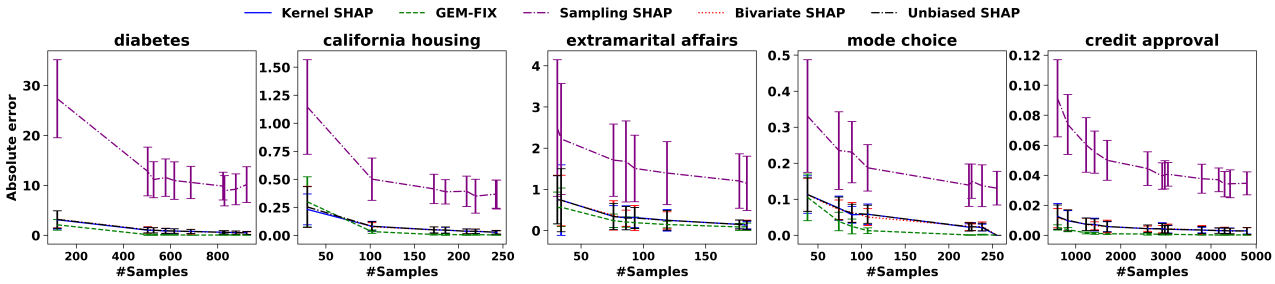

Figure 3: The deviation from ground-truth Shapley value on five tabular datasets.

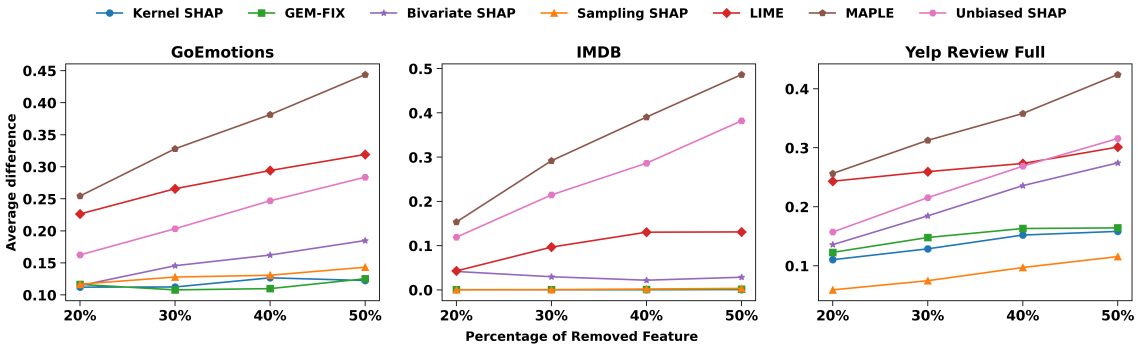

Figure 4: Deviation from original prediction by masking the deemed redundant features.

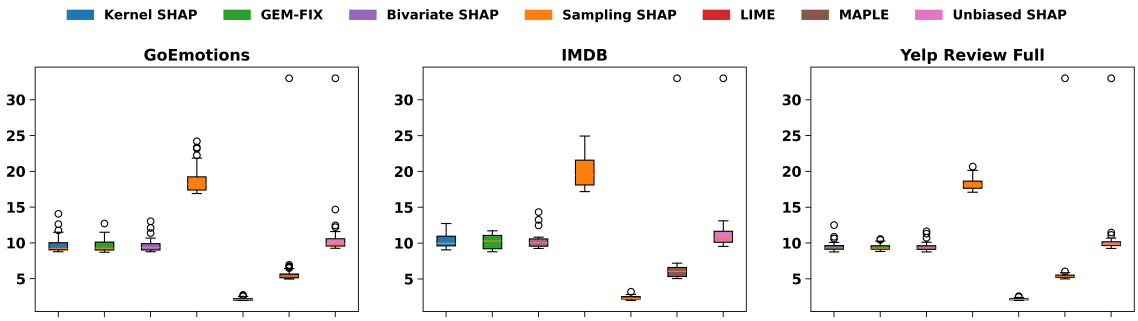

Figure 5: The execution time of methods on explaining the sentiment analysis of 50 reviews.

7 Conclusion and Discussion

GEM-FIX puts forward a novel approach to compute the Shapley value and identify significant interactions in explanation games using their Möbius representation. Unlike traditional estimators, GEM-FIX relaxes the essentiality of games for providing explanations. It addresses the challenge of managing exponentially many parameters of games through a data-driven approach. Initially, an optimization model is efficiently solved, and subsequently, a new Shapley value formulation is developed that limits the complexity to a maximum of $|\mathfrak{S}|$ parameters, where $\mathfrak{S}$ represents the sample set. The method is able to identify only the significant interactions computed from exponentially many cases.

For future research, several paths can be explored. While GEM-FIX effectively identifies significant interactions, the process can become time-intensive with an increase in the number of these interactions. Given the focus on explainability on the most important rather than all significant interactions, further refinement is needed. Future work could include a detailed analysis of the impact of regularization techniques discussed in Section 4.2 on pinpointing these critical interactions. Moreover, GEM-FIX does not clarify the uncertainty associated with Shapley value estimates or the optimal sample sizes required. Integrating Bayesian models could enhance understanding of the estimation uncertainty and aid in determining the necessary sample sizes for accurate estimation.

Acknowledgments

This publication is part of the project PROXCAI, which is financed by the Dutch Research Council (NWO) under the grant *KIVI.2019.003*.

References

Bazaraa, M. S.; Sherali, H. D.; and Shetty, C. M. 2013. *Nonlinear programming: theory and algorithms*. John wiley & sons.

Breuer, N. O.; Sauter, A.; Mohammadi, M.; and Acar, E. 2024. CAGE: Causality-Aware Shapley Value for Global Explanations. In *World Conference on Explainable Artificial Intelligence*, 143–162. Springer.

Chen, H.; Covert, I. C.; Lundberg, S. M.; and Lee, S.-I. 2023. Algorithms to estimate Shapley value feature attributions. *Nature Machine Intelligence*, 5(6): 590–601.

Chen, J.; Song, L.; Wainwright, M.; and Jordan, M. 2018a. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, 883–892. PMLR.

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

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

Covert, I.; Lundberg, S. M.; and Lee, S.-I. 2020. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33: 17212–17223.

Datta, A.; Sen, S.; and Zick, Y. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, 598–617. IEEE.

Fumagalli, F.; Muschalik, M.; Kolpaczki, P.; Hüllermeier, E.; and Hammer, B. 2024. Shap-iq: Unified approximation of any-order shapley interactions. *Advances in Neural Information Processing Systems*, 36.

Grabisch, M. 1996. The representation of importance and interaction of features by fuzzy measures. *Pattern Recognition Letters*, 17(6): 567–575.

Grabisch, M.; and Roubens, M. 1999. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28: 547–565.

Kumar, I.; Scheidegger, C.; Venkatasubramanian, S.; and Friedler, S. 2021. Shapley Residuals: Quantifying the limits of the Shapley value for explanations. *Advances in Neural Information Processing Systems*, 34: 26598–26608.

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

Masoomi, A.; Hill, D.; Xu, Z.; Hersh, C. P.; Silverman, E. K.; Castaldi, P. J.; Ioannidis, S.; and Dy, J. 2021. Explanations of Black-Box Models based on Directional Feature Interactions. In *International Conference on Learning Representations*.

Mohammadi, M.; Tiddi, I.; and Ten Teije, A. 2023. A Local Non-Additive Framework for Explaining Black-Box Predictive Models. In *ECAI 2023*, 1728–1738. IOS Press.

Plumb, G.; Molitor, D.; and Talwalkar, A. S. 2018. Model agnostic supervised local explanations. *Advances in neural information processing systems*, 31.

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

Shapley, L. S. 1953. A value for n-person games.

Strang, G. 2012. *Linear algebra and its applications*.

Štrumbelj, E.; and Kononenko, I. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41: 647–665.

Sundararajan, M.; Dhamdhere, K.; and Agarwal, A. 2020. The shapley taylor interaction index. In *International conference on machine learning*, 9259–9268. PMLR.

Tsai, C.-P.; Yeh, C.-K.; and Ravikumar, P. 2023. Faith-shap: The faithful shapley interaction index. *Journal of Machine Learning Research*, 24(94): 1–42.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.