GENERALIZED PREF-SHAP TO EXPLAIN PREFERENCE FUNCTIONS

Anonymous authors

Paper under double-blind review

ABSTRACT

We address the problem of feature attribution for skew-symmetric preference functions in dueling data settings, using the cooperative game-theoretic concept of *Shapley values*. Building on Pref-SHAP Hu et al. (2022), we propose *Generalized Pref-SHAP*, a framework that extends its applicability to a broader class of preference functions. Our method leverages a simple neural network to model arbitrary feature mappings while exploiting the canonical block structure inherent to skew-symmetric functions, enabling more meaningful explanations. Additionally, we explore foundational questions about Pref-SHAP, including its relationship with the block decomposition structure of generalized preference functions. We perform experiments on a range of synthetic datasets to demonstrate the effectiveness and efficiency of our approach.

1 Introduction

Pairwise preference learning, often modeled via dueling data, plays a central role in ranking, recommendation, and decision-making systems. A common framework for modeling such data involves skew-symmetric preference functions, where the function f(u,v) satisfies f(u,v) = -f(v,u), representing the preference for item u over item v. We can represent such functions using the following canonical form:

$$f(u,v) = u^{\top} A v, \tag{1}$$

where d is even, and $A \in \mathbb{R}^{d \times d}$ is a block-diagonal matrix composed of d/2 skew-symmetric 2×2 rotation matrices $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. This bilinear structure naturally arises in modeling utilities and comparative d is a structure naturally arises.

isons, and is particularly amenable to analysis due to its algebraic simplicity and geometric structure. While modeling such functions has received considerable attention, explaining their predictions, that is, attributing the model's output to individual input features, remains less explored. Recent work, notably Pref-SHAP Hu et al. (2022), extends the classical Shapley value Shapley et al. (1953) to the setting of pairwise comparisons, assigning feature-level attributions to preference decisions using a game-theoretic lens. We aim to investigate the theoretical and practical behavior of Pref-SHAP in models based on the canonical form.

1.1 MAIN CONTRIBUTIONS

- **Block Structure Consistency:** We analyze whether Pref-SHAP respects the block structure inherent in the canonical form. This reveals how well the Shapley attributions align with the inherent feature pairing in the model.
- Generalization to Arbitrary Feature Mappings: We extend Pref-SHAP to work with nonlinear or learned feature maps ϕ , proposing *Generalized Pref-SHAP*. This new framework preserves the interpretability of Shapley decompositions while leveraging the structure in A, enabling explanations in complex models with deep representations.
- Theoretical Analysis Under Canonical Form: We derive closed-form expressions for Pref-SHAP values when f(u, v) = u^TAv, focusing on the two-feature case for clarity. We explore how the attributions are affected by distributional properties like feature variance, constancy, and independence. Our analysis identifies unintuitive behaviors that arise from interaction effects and the symmetry axiom in Shapley values, and suggests that interactionaware explanations may be necessary in some cases.

2 BACKGROUND MATERIALS

2.1 Preference Learning

We consider a more general class of skew-symmetric functions of the form:

$$f(u,v) = \langle \phi(u), A\phi(v) \rangle, \tag{2}$$

where $u,v\in\mathbb{R}^k$, $\phi:\mathbb{R}^k\to\mathbb{R}^d$ is a feature mapping(possibly nonlinear) with even d, and $A\in\mathbb{R}^{d\times d}$. Any skew-symmetric function can be represented in this form, where ϕ is the identity function and k=d. The likelihood function used for such models, especially in pairwise preference scenarios, often takes the form:

$$p(y \mid u, v) = \sigma(y \cdot f(u, v)) = 1 - p(y \mid v, u),$$
 (3)

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function and $y \in \{-1,1\}$ indicates the preference label.

2.2 Shapley Value

The purpose of this paper is to study the local explainability of predictions based on preference functions using the *Shapley value* Shapley et al. (1953), a concept from cooperative game theory. It is basically a credit allocation method where a subset of k players is assigned a value function based on their contribution to a game $\nu:[0,1]^k\to\mathbb{R}$. The Shapley value for player j in game ν is given by:

$$\Phi_{j}(\nu) = \sum_{S \subseteq \Omega \setminus \{j\}} \frac{|S|!(d-|S|-1)!}{d!} \left[\nu(S \cup \{j\}) - \nu(S) \right],$$

where, $\Omega = \{1, \dots, d\}$ is the set of d players.

It satisfies several desirable uniqueness axioms such as efficiency, symmetry, null player property, linearity. The use of this concept in explainable ranking context helps in finding feature attribution in a model prediction by creating a correspondence between the concept of players and item features. In particular, linearity ensures that for a linear ensemble of models, the Shapley value of a feature is the corresponding linear combination of individual model Shapley values.

2.3 PREF-SHAP

In the Pref-SHAP framework Hu et al. (2022), the value function is adapted to model the pairwise preference setting.

2.4 Preferential Value Function for Items Hu et al. (2022); Chau et al. (2022b)

Definition 1. Given a preference function $f \in \mathcal{H}$, and a pair of items $(x^l, x^r) \in \mathcal{X} \times \mathcal{X}, \mathcal{X} \subseteq \mathcal{R}^d$, the preferential value function $\nu : \mathcal{X} \times \mathcal{X} \times [0,1]^k \times \mathcal{H} \to \mathbb{R}$ for computing Shapley values in Pref-SHAP is defined as the following conditional expectation:

$$\nu_{x^l, x^r, S}(f) = \mathbb{E}_r \left[f(\{X_S^l, X_{S^c}^l\}, \{X_S^r, X_{S^c}^r\}) \middle| X_S^l = x_S^l, X_S^r = x_S^r \right], \tag{4}$$

where $S \subseteq \{1, ..., k\}$ is a subset of feature indices, and X is the input random vector. The notation X_S denotes the subvector of features indexed by S, and S^c is the complement of S. The pair $\{X_S, X_{S^c}\}$ refers to the full input vector X formed by concatenation. The reference distribution r is defined as:

$$r = r(X_{S^c}^l, X_{S^c}^r \mid X_S^l = x_S^l, X_S^r = x_S^r).$$

2.5 GENERALIZED PREFERENTIAL KERNELCHAU ET AL. (2022A)

Definition 2. Given a kernel $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, defined on the original feature space $\mathcal{X} \subseteq \mathcal{R}^d$, the Generalized Preferential Kernel k_E is defined as:

$$k_E((x_i^l, x_i^r), (x_i^l, x_i^r)) = k(x_i^l, x_i^l) \cdot k(x_i^r, x_i^r) - k(x_i^l, x_i^r) \cdot k(x_i^r, x_i^l),$$
(5)

where the skew-symmetric function f is assumed to lie in the Reproducing Kernel Hilbert Space (RKHS) \mathcal{H}_{k_E} associated with k_E .

2.6 BLOCK STRUCTURE OF SHAPLEY VALUES

One of the key properties satisfied by Shapley values, and more importantly for our purposes by Pref-Shap, is the *efficiency* axiom which states that the sum of all feature attributions equals the overall preference score. In particular, for a feature space of dimension d where features are grouped into d/2 disjoint consecutive blocks of size 2, we have:

$$\sum_{i=1}^{d/2} (\Phi_{2i-1} + \Phi_{2i}) = \sum_{i=1}^{d/2} (u_{2i-1}v_{2i} - u_{2i}v_{2i-1}) = u^{\top} A v,$$
 (6)

where, d is even, and $A \in \mathbb{R}^{d \times d}$ is a block-diagonal matrix composed of d/2 skew-symmetric 2×2 rotation matrices $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ as defined in 1. Each block corresponds to an antisymmetric interaction between two consecutive features. A natural question that arises is whether Pref-SHAP also satisfies a finer-grained block decomposition property at the level of individual 2-dimensional feature blocks.

That is, for each odd index
$$i \in \{1, 3, \dots, d-1\}$$
, does the following hold?
$$\Phi_i + \Phi_{i+1} = u_i v_{i+1} - u_{i+1} v_i = u^\top A_{i:i+1} v, \text{ for } i \bmod 2 \neq 0,$$
 (7)

where $A_{i:i+1} \in \mathbb{R}^{d \times d}$ is a matrix whose only nonzero entries lie in a 2×2 skew-symmetric submatrix spanning rows and columns i and i+1, and zeros elsewhere. Thus, the full matrix A can be written as a sum of these block-local matrices: $A = \sum_{i=1}^{d/2} A_{2i-1:2i}$. Here, $A_{2i-1:2i}$ denotes the 2×2 submatrix of A corresponding to rows and columns 2i-1 and 2i. This structure leads us to ask: under what conditions does Pref-SHAP decompose additively over such blocks, preserving the local attribution property in Eq. equation 7?

3 Does Pref-Shap Obey the Block Pattern?

Proposition 1 (Block Decomposition of Conditional Pref-SHAP under Independence). Consider a preference function $f: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined on feature vectors $u, v \in \mathbb{R}^d$, where the d features are partitioned into d/2 disjoint consecutive blocks: $B_j = \{2j-1, 2j\}, \quad j=1, \ldots, \frac{d}{2}$.

Assume one of the following:

- Full independence: All features $\{X_1, \ldots, X_d\}$ are mutually independent, or
- Blockwise independence: Features within each block B_j may be dependent, but blocks are mutually independent, i.e., $X_{B_i} \perp X_{B_j}$ for all $i \neq j$, where X_{B_j} denotes the features in block B_j .

Suppose, the preference function decomposes additively over blocks: $f(u, v) = \sum_{j=1}^{d/2} f_j(u_{B_j}, v_{B_j})$, where each f_j depends only on features in block B_j .

Pref-SHAP value for feature $i \in [d]$ is analytically computed as

$$\Phi_i := \sum_{S \subseteq [d] \setminus \{i\}} \frac{|S|!(d-1-|S|)!}{d!} \left[\nu(S \cup \{i\}) - \nu(S) \right],$$

where the conditional value function is, $\nu(S) := \mathbb{E}[f(u,v) \mid (u_k, v_k)_{k \in S}].$

Then, under the above assumptions, $\sum_{i \in B_j} \Phi_i = f_j(u_{B_j}, v_{B_j})$, for all $j = 1, \dots, \frac{d}{2}$. Moreover, the individual Pref-SHAP values Φ_i generally depend on the full feature set due to the global conditioning in ν , but their sum over each block recovers the exact blockwise preference contribution. If the features are correlated across blocks (i.e., the blockwise independence assumption fails), then this additive decomposition of the Pref-SHAP values does not generally hold.

Proof Sketch. Consider features partitioned into blocks B_1, \ldots, B_m . Pref-SHAP values are computed via differences of conditional expectations. Under full or blockwise independence, conditional expectations of features outside a block reduce to constants independent of conditioning subsets.

Consequently, cross-block terms appear symmetrically in the Shapley difference terms and cancel out. This implies that Pref-SHAP values decompose additively over blocks, with each block's attribution depending only on its own features. In contrast, when features are arbitrarily correlated across blocks, these cancellations no longer occur, and the decomposition fails. The complete proof is provided in the appendix.

3.1 CAN SHAPLEY VARIANTS RECOVER BLOCK PATTERN?

This discussion and the related literature are included in the appendix due to space constraint.

3.2 CAN WE LEARN A FEATURE MAPPING TO RESTORE BLOCK STRUCTURE?

We investigate whether it is possible to map the original features to a transformed space—such as an eigenspace defined by the covariance matrix—so that the PREF-SHAP values exhibit a block structure in the transformed coordinates. Specifically, we consider an orthonormal linear transformation $W \in \mathbb{R}^{d \times d}$ such that $x = W^T u$, $y = W^T v$, where $u, v \in \mathbb{R}^d$ are original feature vectors, and

$$x,y \in \mathbb{R}^d \text{ are transformed features. } W = \begin{bmatrix} w_{11} & w_{21} & \cdots & w_{d1} \\ w_{12} & w_{22} & \cdots & w_{d2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1d} & w_{2d} & \cdots & w_{dd} \end{bmatrix}, u = Wx, \quad v = Wy \text{ are the}$$

inverse transformations.

Each transformed feature is, $x_i = \sum_{j=1}^d w_{ji} u_j, \quad y_i = \sum_{j=1}^d w_{ji} v_j, \quad \forall i = 1, \dots, d.$

Example: Skew-Symmetric Function with 4 Features Consider the skew-symmetric preference function $f(u, v) = u_1v_2 - u_2v_1 + u_3v_4 - u_4v_3$, which exhibits a natural block structure with two blocks: $\{1, 2\}$ and $\{3, 4\}$.

Expressing f in the transformed space,

$$f(u,v) = f(Wx, Wy) = (Wx)_1(Wy)_2 - (Wx)_2(Wy)_1 + (Wx)_3(Wy)_4 - (Wx)_4(Wy)_3$$
$$= \sum_{i,j=1}^d (w_{i1}w_{j2} - w_{i2}w_{j1} + w_{i3}w_{j4} - w_{i4}w_{j3}) (x_iy_j - x_jy_i).$$

Rearranging terms into differences of products (to maintain skew-symmetry), $f(x,y) = \sum_{i < j} c_{ij} (x_i y_j - x_j y_i)$, where, $c_{ij} = w_{i1} w_{j2} - w_{i2} w_{j1} + w_{i3} w_{j4} - w_{i4} w_{j3}$.

Due to the linear mixing by W, the original block structure (only interactions within blocks $\{1,2\}$ and $\{3,4\}$) generally disappears. Instead, f becomes a fully connected skew-symmetric form involving all pairs (i,j) of transformed features, where all c_{ij} can be nonzero. Hence, the model in the transformed space no longer decomposes into independent blocks. Consequently, the PREF-SHAP values computed on transformed features will not obey the block pattern seen in the original features. This observation generalizes to higher dimensions and nonlinear models with interaction terms: the orthonormal transform W mixes all features, distributing interactions across transformed features. Nonlinear terms (e.g., products of features from different blocks) are generally not diagonalized by W. Therefore, block independence is a property of the function structure, not solely the feature covariance. Shapley values are generally not linear in the features or their transformations, except for linear models. Thus, for a linear model $f(u) = \beta^T u$, transforming features linearly as $x = W^T u$ yields $f(u) = \beta^T u = \beta^T W x = \tilde{\beta}^T x$, and Shapley values transform linearly: $\phi_{u_j} = \sum_i w_{ji} \phi_{x_i}$. For nonlinear models with feature interactions, this linearity breaks, and Shapley values of original features cannot be represented as linear combinations of transformed features' Shapley values.

Remark: Mapping features to an eigenspace or another orthonormal basis does not, in general, preserve the block independence structure of nonlinear interaction models like skew-symmetric preference functions. Consequently, PREF-SHAP values computed in the transformed space will reflect fully coupled feature interactions and fail to obey the original block pattern. Only in the special case of linear models, such a transformation preserve functional structure and yield a linear relation between original and transformed features' Shapley values.

4 STUDY OF PREF-SHAP PROPERTIES W.R.T THE CANONICAL FORM

When there are only two features, the expression for Pref-SHAP reduces to:

$$\Phi_{1} = \frac{1}{2} [(u_{1}v_{2} - u_{2}v_{1}) + u_{1}\mathbb{E}[Y_{2}|Y_{1} = v_{1}] - v_{1}\mathbb{E}[X_{2}|X_{1} = u_{1}]
- v_{2}\mathbb{E}[X_{1}|X_{2} = u_{2}] + u_{2}\mathbb{E}[Y_{1}|Y_{2} = v_{2}] - \mathbb{E}[X_{1}]\mathbb{E}[Y_{2}] + \mathbb{E}[Y_{1}]\mathbb{E}[X_{2}]]
\Phi_{2} = \frac{1}{2} [(u_{1}v_{2} - u_{2}v_{1}) - u_{1}\mathbb{E}[Y_{2}|Y_{1} = v_{1}] + v_{1}\mathbb{E}[X_{2}|X_{1} = u_{1}]
+ v_{2}\mathbb{E}[X_{1}|X_{2} = u_{2}] - u_{2}\mathbb{E}[Y_{1}|Y_{2} = v_{2}] - \mathbb{E}[X_{1}]\mathbb{E}[Y_{2}] + \mathbb{E}[Y_{1}]\mathbb{E}[X_{2}]]$$
(8)

Since $\mathbb{E}(X_i) = \mathbb{E}(Y_i) \ \forall i \in \{1, 2\}$, the last two terms vanish.

4.1 IMPACT OF FEATURE VARIANCE ON PREF-SHAP

Let us consider this analytically. Suppose the features are independent, then the Pref-SHAP values simplify to:

$$\Phi_{1} = \frac{1}{2} \left[(u_{1}v_{2} - u_{2}v_{1}) + (u_{1}\mathbb{E}(Y_{2})) - (v_{1}\mathbb{E}(X_{2})) - (v_{2}\mathbb{E}(X_{1})) + (u_{2}\mathbb{E}(Y_{1})) \right]
\Phi_{2} = \frac{1}{2} \left[(u_{1}v_{2} - u_{2}v_{1}) - (u_{1}\mathbb{E}(Y_{2})) + (v_{1}\mathbb{E}(X_{2})) + (v_{2}\mathbb{E}(X_{1})) - (u_{2}\mathbb{E}(Y_{1})) \right]$$
(9)

Now, suppose, $\mathrm{Var}(X_1) = \mathrm{Var}(Y_1) = 100$ and $\mathrm{Var}(X_2) = \mathrm{Var}(Y_2) = 1$. Consider an instance where $u_1 = 100$, $v_1 = -100$, $u_2 = 1$, $v_2 = -1$. In this case, the first-order terms involving u_1 and v_1 dominate, so Φ_1 could be significantly larger than Φ_2 . However, for another instance with opposite signs (e.g., $u_1 = -100$, $v_1 = 100$, $v_2 = -1$, $v_2 = 1$), $v_3 = 100$, $v_4 = 100$, $v_5 = 100$, $v_7 = 100$,

Therefore, while individual Pref-SHAP values may be affected by the variance in the features, this effect can cancel out when averaging over many samples. In other words, the global (dataset-level) Pref-SHAP attributions are not directly biased by feature variance in expectation under independence. Hence, Pref-SHAP with conditional value functions represents the true conditional expectation of the model output. Thus, it does not directly encode feature variance. However, the uncertainty in prediction caused by high feature variance (especially in correlated settings) may influence the attribution in an indirect way. This distinction aligns with insights from information-theoretic approaches to attribution Watson et al. (2024), where variance influences model uncertainty but not necessarily marginal attributions unless the model or attribution method explicitly encodes that dependency.

4.2 EFFECT OF CONSTANT FEATURES ON PREF-SHAP

The full discussion with analysis is deferred to the appendix due to space constraints. Briefly, even if a feature is constant across item pairs, it may still receive non-zero attribution under Pref-SHAP feature correlations. However, under certain independence assumptions and symmetry axiom, such features may yield equal attributions due to model structure.

Case	Baseline Form	Φ_1	Notes
1	$z = \frac{u+v}{2}$	$\frac{1}{2}(u_1v_2 - u_2v_1 + u_1u_2 - v_1v_2)$	Pair-specific baseline
2	$z = \mathbb{E}[Z]$	$\frac{1}{2}(u_1v_2 - u_2v_1 + (u_1 - v_1)\mathbb{E}[Z_2] - (v_2 - u_2)\mathbb{E}[Z_1])$	Global baseline

Figure 1: Comparison of Baseline Shapley Forms

4.3 Pref-SHAP vs. Baseline Shapley Sundararajan & Najmi (2020)

Baseline
$$z = (z_1, z_2) = \left(\frac{u_1 + v_1 + \dots}{n}, \frac{u_2 + v_2 + \dots}{n}\right) = \left(\mathbb{E}[Z_1], \mathbb{E}[Z_2]\right)$$
 (10)

In Case 1(Fig. 1), when features are equal, attribution vanishes. In Case 2, even equal features may have non-zero attribution if they deviate from the global baseline $\mathbb{E}[Z]$. Pref-SHAP resembles Case 2 and better accounts for meaningful deviations from population averages.

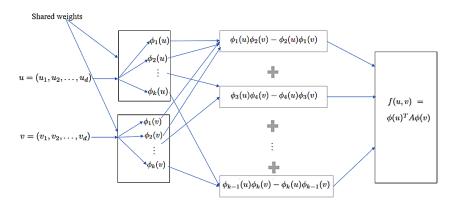


Figure 2: Generalized Pref-SHAP architecture using a simple neural network

Comparison of Feature Importance Methods Spearman IG:BO=1.00, IG:BN=1.00, IG:PO=0.60 | Kendall IG:BO=1.00, IG:BN=1.00, IG:PO=0.33

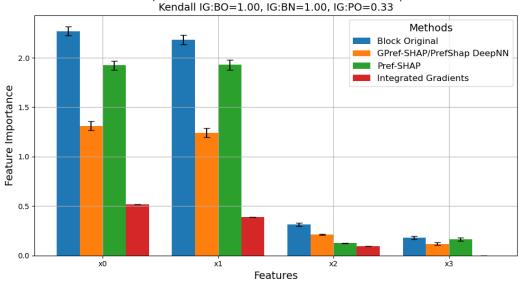


Figure 3: Quadratic feature mappings $\{x_0^2, x_1^2, x_0x_1, x_0x_2\}$. Generalized Pref-SHAP is abbreviated as **GPref-SHAP**; **IG** = Integrated Gradients, **BO** = Block Original, **BN** = Block DeepNN, and **PO** = Pref-SHAP Original. **IG:BO** means the rank correlation between them. The terms **Pref-SHAP DeepNN** and **Block DeepNN** are used interchangeably to refer to GPref-SHAP. Terms in the captions represent the mapped features.

5 GENERALIZED PREFERENTIAL KERNEL INSIGHT:

Given the bilinear form of the Generalized Preferential kernelChau et al. (2022a):

$$\sum_{i=1}^{n} \alpha_i K_E((x_i, y_i), (x_{\text{test}}, y_{\text{test}})) = \sum_{i=1}^{n} \alpha_i \left[(x_i^\top x_{\text{test}}) (y_i^\top y_{\text{test}}) - (x_i^\top y_{\text{test}}) (y_i^\top x_{\text{test}}) \right] = x_{\text{test}}^\top \left(\sum_{i=1}^{n} \alpha_i (x_i y_i^\top - y_i x_i^\top) \right) y_{\text{test}}.$$

the model output becomes a bilinear form over input pairs with weights represented by a skew-symmetric matrix i.e. $\sum_{i=1}^{n} \alpha_i(x_iy_i^{\top} - y_ix_i^{\top})$ in a transformed space. This naturally aligns with the linear kernel as the base kernel (instead of RBF kernel), which is both expressive and computationally efficient in this context, i.e. in cases where the feature map ϕ is the identity function.

Effective Feature Space for Pref-SHAP. In this formulation, the prediction function is linear in the space of effective features, which are formed as pairwise interactions of the original features. As each pair of features (x_i, x_j) contributes an interaction term, the number of effective features becomes $\binom{d}{2}$, significantly expanding the representational capacity. The final prediction is thus: $f(u,v) = \sum_{i < j} \alpha_{ij} (u_i v_j - u_j v_i)$, where α_{ij} are the learned coefficients associated with each interaction. This formulation allows us to attribute importance to each original feature based on the strength and frequency of its interactions across data instances.

6 PROPOSED MODEL: GENERALIZED PREF-SHAP

Algorithm 1 Generalized Pref-SHAP

Input: Pairwise item data $\{X_l, X_r, Y\}$ with features $X \in \mathbb{R}^d$, number of blocks k for feature mapping.

- 1. Learn the unknown feature mapping $\phi = (\phi_1, \phi_2, \dots, \phi_k)$ using the neural network shown in Figure 6, where the predicted output is: $f(x_l, x_r) = \phi(x_l)^T A \phi(x_r) = \sum_{i=1}^{k/2} (\phi_{2i-1}(x_l)\phi_{2i}(x_r) \phi_{2i}(x_l)\phi_{2i-1}(x_r))$.
- 2. Construct k/2 datasets, each corresponding to a blockwise pairwise interaction of mapped features using ϕ .
- For each block dataset, apply Kernel Ridge Regression (KRR) to learn a component function.
- 4. Compute the residual between the original label Y and the sum of predictions from all k/2 component models. Apply KRR to this residual dataset.
- 5. Compute Pref-SHAP values for each of the k/2 blocks and the residual component, giving a $k \times (k/2+1)$ matrix of attributions.
- 6. Aggregate feature attributions by summing across columns (i.e., summing attributions across blocks for each feature).

Output: Final Pref-SHAP value for each original input feature.

In standard Pref-SHAP, the model approximates the skew-symmetric function of the mapped features using Kernel Ridge Regression(KRR) but does not learn the mapping ϕ explicitly. Consequently, it cannot exploit the canonical block structure inherent to skew-symmetric functions. In contrast, our proposed *Generalized Pref-SHAP* explicitly learns the feature mapping ϕ , enabling a structured decomposition of the preference function into interpretable blocks. This decomposition preserves the block structure of the underlying function and facilitates more accurate and meaningful feature attributions. By jointly learning the feature representation and maintaining block-wise interpretability, *Generalized Pref-SHAP* improves the transparency and faithfulness of the attribution process, aligning explanations more closely with the structure of the learned model.

The network takes as input a pair of items represented using their original features and learns the feature mappings of each item via weight sharing. The learned mappings are then passed into a module which computes the skew-symmetric preference function for these mapped features. The detailed explanation of the algorithmic steps are described in appendix A.

7 EXPERIMENTS

In this section, we describe our experimental setup and results on both synthetic and real-world datasets. We have conducted experiments mainly on carefully generated synthetic data because the motivation behind Generalized Pref-SHAP is rooted in the design of feature mappings ϕ , and real-world datasets rarely provide a ground truth for global feature importance. However, in domains where expert knowledge about feature relevance exists, real-data experiments can help identify the more interpretable model. It is quite possible that the explicit features in the real data get mapped to some hidden space before applying the skew-symmetric preference function instead of directly contributing to the preferences and this phenomenon can be better explained through our proposed

method. We have used the publicly available dataset *Pokemon*(Nguyen Van Anh (2021)) for our real-world experiment and the related result (bar plot) is deferred to the appendix.

7.1 SYNTHETIC EXPERIMENTS

We conduct experiments using synthetic data consisting of n=100 items, where each item's features are sampled from a 4-dimensional Gaussian distribution with mean zero and identity covariance: $x_i \sim \mathcal{N}(0, \mathbf{I})$ for $i=1,\ldots,100$. We construct preference labels using various differentiable feature mappings ϕ , including:

Polynomial mappings: quadratic expansions, **Sinusoidal mappings:** combinations of $\sin(\cdot)$, $\cos(\cdot)$ applied to linear projections, **Hybrid mappings:** combinations of polynomial and sinusoidal transformations, **Neural network mappings:** small feedforward networks with ReLU activations and varying depth/width.

The preference labels are generated by applying the skew-symmetric function f to transformed features. We evaluate the feature importance scores computed using four methods: **Pref-SHAP**: the original method using conditional Shapley values on learned KRR models. **Generalized Pref-SHAP** (**Ours**): learns each skew-symmetric component block via neural networks, then aggregates their Shapley values. **Integrated Gradients** (**Baseline**): standard attribution baseline for differentiable models. **Block-Original**: a reference method that trains separate KRR models per block and aggregates block-level Pref-SHAP values.

For the ground truth, we compute the global feature importance scores with respect to the feature mapping function using *Integrated Gradients*, and then propagate those scores to the canonical skewsymmetric preference function. Since each block in the canonical form contains exactly two terms, with each feature appearing once per term, the importance score of each feature is doubled per block. As the blocks are disjoint, the final feature importance is obtained by summing these contributions across blocks. This enables comparison using rank-based metrics. We use all $\binom{100}{2}$ unique pairs for training and evaluation, as the task is regression-based whereas our real-world experiment is based on classification of pairwise preferences. The pairwise comparisons with labels are split into training (80%), validation (10%), and test (10%) sets and the experiments are averaged over 5 folds. Feature attribution is computed on a randomly selected subset of the pairwise data. For Kernel Ridge Regression, hyperparameter tuning is performed via gradient-based optimization using the Falkon library. For the neural network setup, we fix two hidden layers and employ Bayesian optimization using Optuna with 100 trials to tune the hyperparameters: the number of hidden nodes (shared across layers) is selected from the set {32,64,96,128}, while the learning rate and weight decay are sampled from logarithmic ranges $[10^{-4}, 10^{-1}]$ and $[10^{-6}, 10^{-2}]$ respectively. We use the mean squared error (MSE) loss and optimize using the Adam optimizer with a batch size of 64 and early stopping based on validation loss.

7.2 EVALUATION METRICS

We report three types of plots for each setup: **Bar plots:** Global feature importance scores (averaged absolute attribution values across test samples) for each method. *Spearman and Kendall Tau rank correlations* between each method's global feature importance scores and the ground truth (or baseline Integrated Gradients) are computed. **Linearity plots:** Scatterplots comparing predicted Pref-SHAP attributions to the sum of block-wise learned component attributions(Block-Original, Generalized Pref-SHAP), validating linearity. **RMSE curves:** Test error convergence (RMSE) across optimization iterations in the KRR model. As the Falkon library used in Pref-SHAP uses a gradient descent based hyperparameter optimization, hence for each iteration, the plot represents RMSE value between the original skew-symmetric labels and the learned labels for each algorithm. **Sanity check:** In cases where only a subset of features is used to generate the label, we verify that inactive features are assigned near-zero importance. The corresponding plots are provided in the appendix.

7.3 Observations

Generalized Pref-SHAP consistently produces more meaningful and interpretable global feature importance scores than Pref-SHAP, particularly when the true feature importance is sparse or block-structured. Linearity plots show that both Pref-SHAP and our method yield highly correlated local

 feature attributions, verifying the block-decomposability of attributions in practice. RMSE plots show that Generalized Pref-SHAP models achieve comparable or better function approximation performance relative to Pref-SHAP and Block-Original. Overall, the experiments demonstrate that Generalized Pref-SHAP is more aligned with ground-truth structure in synthetic data and offers improved interpretability while maintaining the theoretical properties of the original method.

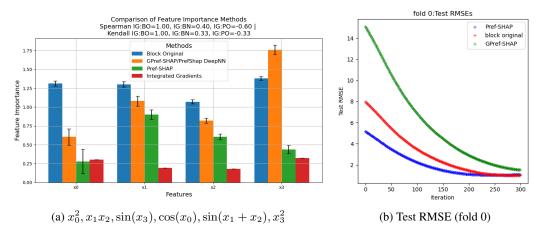
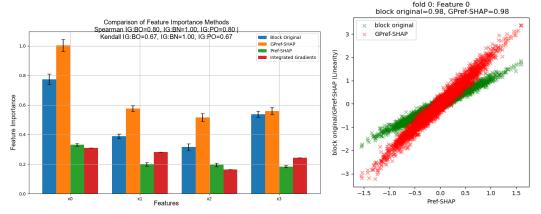


Figure 4: (a)hybrid mapping of features, (b) ReLU network (4 input features, 6 mapped features, 4 hidden layers, 16 nodes in each layer).



(a) ReLU network (4 input features, 6 mapped features, 4 hidden (b) Linearity of Pref-SHAP variants (fold layers, 16 nodes in each layer) 0, feature x_0

Figure 5: ReLU network

8 CONCLUSION

In this work, we introduced *Generalized Pref-SHAP*, a principled extension of Pref-SHAP designed to provide faithful feature attributions for pairwise preference models that involve rich, nonlinear feature mappings. We analyzed the theoretical properties of Pref-SHAP and showed that under independence or blockwise independence, Pref-SHAP decomposes additively over feature blocks, highlighting its interpretability in structured models. Our synthetic experiments demonstrate that Generalized Pref-SHAP recovers global feature importance more accurately, aligns better with ground-truth structure, and satisfies desirable attribution properties such as linearity and sparsity in inactive features whereas the real data experiment shows that *Generalized Pref-SHAP* has comparable performance with Pref-SHAP while explaining the important features for the paiwise preferences.

REFERENCES

- Siu Lun Chau, Javier Gonzalez, and Dino Sejdinovic. Learning inconsistent preferences with gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 2266–2281. PMLR, 2022a.
- Siu Lun Chau, Robert Hu, Javier Gonzalez, and Dino Sejdinovic. Rkhs-shap: Shapley values for kernel methods. *Advances in neural information processing systems*, 35:13050–13063, 2022b.
- Fabian Fumagalli, Maximilian Muschalik, Patrick Kolpaczki, Eyke Hüllermeier, and Barbara Hammer. Kernelshap-iq: Weighted least-square optimization for shapley interactions. *arXiv* preprint *arXiv*:2405.10852, 2024.
- Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *Advances in neural information processing systems*, 33:4778–4789, 2020.
- Robert Hu, Siu Lun Chau, Jaime Ferrando Huertas, and Dino Sejdinovic. Explaining preferences with shapley values. *Advances in Neural Information Processing Systems*, 35:27664–27677, 2022.
- Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable ai: A causal problem. In *International Conference on artificial intelligence and statistics*, pp. 2907–2916. PMLR, 2020.
- Tuan Nguyen Van Anh. Pokémon dataset with team combat. https://www.kaggle.com/datasets/tuannguyenvananh/pokemon-dataset-with-team-combat, 2021. Accessed: 2025-09-23.
- Lloyd S Shapley et al. A value for n-person games. 1953.
- Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186, 2020.
- Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pp. 9269–9278. PMLR, 2020.
- Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *International conference on machine learning*, pp. 9259–9268. PMLR, 2020.
- Muhammad Faaiz Taufiq, Patrick Blöbaum, and Lenon Minorics. Manifold restricted interventional shapley values. In *International Conference on Artificial Intelligence and Statistics*, pp. 5079–5106. PMLR, 2023.
- David Watson, Joshua O'Hara, Niek Tax, Richard Mudd, and Ido Guy. Explaining predictive uncertainty with information theoretic shapley values. *Advances in Neural Information Processing Systems*, 36, 2024.

A APPENDIX

Detailed Explanation of the Algorithm Steps:

- 1. Learning the Feature Mapping ϕ : The algorithm begins by learning a feature mapping $\phi = (\phi_1, \phi_2, \dots, \phi_k)$ via a neural network architecture (illustrated in Figure 6). This network takes as input a pair of items (x_l, x_r) with original features in \mathbb{R}^d , and outputs their transformed features. The learned mapping ϕ is structured to reflect the block-wise decomposition, where each pair (ϕ_{2i-1}, ϕ_{2i}) corresponds to a canonical 2×2 skew-symmetric block. This decomposition explicitly encodes the skew-symmetric structure in the feature space.
- 2. Constructing Blockwise Datasets: After learning the feature mappings, the dataset is partitioned into k/2 separate block datasets, each corresponding to the pairwise interaction between the mapped features ϕ_{2i-1} and ϕ_{2i} . These datasets isolate the contribution of each skew-symmetric block to the overall preference prediction.
- 3. **Fitting Kernel Ridge Regression (KRR) Models:** For each block dataset, Kernel Ridge Regression is applied to learn a component function that models the preference contribution from that specific block. KRR allows flexible, nonparametric fitting that captures the potentially complex relationships within each block.
- 4. **Modeling the Residual:** The sum of predictions from all blockwise KRR models provides a partial reconstruction of the original preference function. This approximation may not capture all nuances of the data, particularly nonlinearities or interactions not aligned with the learned blocks. We compute the residual as the difference between the true labels Y and the aggregated predictions from the blockwise models, thereby capturing information not explained by the canonical blocks. A separate KRR model is then fit to this residual dataset, enabling the method to recover additional signal beyond what is captured by the blockwise decomposition.
- 5. Computing Pref-SHAP Values: Pref-SHAP values are computed individually for each of the k/2 block component models and the residual model. This yields a matrix of attributions of size $k \times (k/2+1)$, where each column corresponds to a block (or residual), and each row corresponds to one of the k mapped features. These attributions represent the contribution of each mapped feature within its block to the preference prediction.
- 6. Aggregating Feature Attributions: Finally, to obtain the attributions for the original input features, the algorithm sums the Pref-SHAP values across all blocks and the residual for each original feature. This aggregation consolidates the blockwise contributions and residual effects into a single attribution score per feature, reflecting the overall importance of each input feature in driving the preference decisions.

B Proposition 1

Proof. We prove the proposition in three steps: base case d=4, generalization to arbitrary even d, and failure of decomposition under correlation.

Step 1: Base case (d = 4)

Partition features into two blocks:

$$B_1 = \{1, 2\}, \quad B_2 = \{3, 4\}.$$

By assumption, the preference function decomposes additively:

$$f(u,v) = f_1(u_{B_1}, v_{B_1}) + f_2(u_{B_2}, v_{B_2}).$$

For feature i = 1, the Pref-SHAP value is

$$\Phi_1 = \sum_{S \subseteq \{2,3,4\}} w(S) \big[\nu(S \cup \{1\}) - \nu(S) \big], \quad w(S) := \frac{|S|!(3 - |S|)!}{4!}.$$

By linearity of expectation,

$$\nu(S) = \mathbb{E}[f(u,v) \mid (u_k, v_k)_{k \in S}] = \mathbb{E}[f_1 \mid S] + \mathbb{E}[f_2 \mid S].$$

Blockwise independence implies that conditioning factorizes:

$$\mathbb{E}[f_j \mid S] = \mathbb{E}[f_j \mid S \cap B_j], \quad j = 1, 2.$$

Since $1 \in B_1$, the marginal contribution satisfies

$$\nu(S \cup \{1\}) - \nu(S) = \left[\nu_1((S \cup \{1\}) \cap B_1) - \nu_1(S \cap B_1)\right] + 0,$$

because $(S \cup \{1\}) \cap B_2 = S \cap B_2$ and f_2 does not change.

Decompose S as $S = S_1 \cup S_2$, where

$$S_1 = S \cap B_1 \subseteq \{2\}, \quad S_2 = S \cap B_2 \subseteq \{3, 4\}.$$

Then,

$$\Phi_1 = \sum_{S_1 \subset \{2\}} \sum_{S_2 \subset \{3,4\}} w(S_1 \cup S_2) \big[\nu_1(S_1 \cup \{1\}) - \nu_1(S_1) \big].$$

The marginal contribution depends only on S_1 , so

$$\Phi_1 = \sum_{S_1 \subseteq \{2\}} \left[\nu_1(S_1 \cup \{1\}) - \nu_1(S_1) \right] \underbrace{\sum_{S_2 \subseteq \{3,4\}} w(S_1 \cup S_2)}_{\text{weight sum}}.$$

By the combinatorial properties of Shapley weights, the inner sum over S_2 equals the Shapley weight on block B_1 :

$$\sum_{S_2 \subseteq \{3,4\}} w(S_1 \cup S_2) = \frac{|S_1|!(|B_1| - 1 - |S_1|)!}{|B_1|!} = \frac{|S_1|!(1 - |S_1|)!}{2!}.$$

Thus,

$$\Phi_1 = \sum_{S_1 \subseteq \{2\}} \frac{|S_1|!(1-|S_1|)!}{2!} \left[\nu_1(S_1 \cup \{1\}) - \nu_1(S_1) \right].$$

Applying the same argument to feature 2, and then summing $\Phi_1 + \Phi_2$ yields

$$\Phi_1 + \Phi_2 = f_1(u_{B_1}, v_{B_1}).$$

Similarly, $\Phi_3 + \Phi_4 = f_2(u_{B_2}, v_{B_2})$.

Step 2: Generalization to arbitrary even *d*

For $i \in B_j$, decompose any subset $S \subseteq [d] \setminus \{i\}$ as

$$S = S_i \cup S_{-i}, \quad S_i \subseteq B_i \setminus \{i\}, \quad S_{-i} \subseteq [d] \setminus (B_i \cup \{i\}).$$

Using linearity,

$$\nu(S) = \sum_{m=1}^{d/2} \nu_m(S \cap B_m).$$

By blockwise independence,

$$\nu_m(S \cap B_m) = \nu_m(S_i \cap B_m)$$
 if $m = j$,

and conditioning on S_{-i} does not affect ν_i .

Hence,

$$\nu(S \cup \{i\}) - \nu(S) = \nu_j(S_j \cup \{i\}) - \nu_j(S_j).$$

Then,

$$\Phi_i = \sum_{S_j \subseteq B_j \setminus \{i\}} \sum_{S_{-j}} w(S_j \cup S_{-j}) \big[\nu_j(S_j \cup \{i\}) - \nu_j(S_j) \big].$$

Marginal contributions depend only on S_i , so summing weights over S_{-i} gives

$$\sum_{S_{-j}} w(S_j \cup S_{-j}) = \frac{|S_j|!(|B_j| - 1 - |S_j|)!}{|B_j|!},$$

the Shapley weight inside block B_i .

Therefore,

$$\Phi_i = \sum_{S_j \subseteq B_j \setminus \{i\}} \frac{|S_j|!(|B_j| - 1 - |S_j|)!}{|B_j|!} \left[\nu_j(S_j \cup \{i\}) - \nu_j(S_j) \right].$$

Summing over all $i \in B_i$,

$$\sum_{i \in B_i} \Phi_i = f_j(u_{B_j}, v_{B_j}).$$

Special Case: Full Independence. When all features are mutually independent (i.e., each feature forms its own block), the block decomposition reduces to:

$$f(u, v) = \sum_{j=1}^{d} f_j(u_j, v_j),$$

with $B_j = \{j\}$. Since each block now contains only one feature, the Pref-SHAP attribution for feature i becomes:

$$\Phi_i = \nu_i(\{i\}) - \nu_i(\emptyset) = f_i(u_i, v_i),$$

because the Shapley value over a singleton block reduces to the full contribution of that feature. Thus, Pref-SHAP values are fully local and additive over individual features, and the global attribution decomposes as:

$$\sum_{i=1}^{d} \Phi_i = f(u, v).$$

This recovers the case of *full additivity* and *local interpretability* under complete feature independence.

Step 3: Necessity and failure under correlation

If features are correlated across blocks, the conditional expectation does not factorize:

$$\mathbb{E}[f_j \mid S] \neq \mathbb{E}[f_j \mid S \cap B_j].$$

Thus,

$$\nu(S \cup \{i\}) - \nu(S)$$

depends on conditioning on features outside B_i .

This breaks the factorization of Shapley weights and the blockwise decomposition of Pref-SHAP values fails.

Remark:

The Shapley value's symmetry axiom ensures that features with identical marginal contributions receive equal attribution, enabling cancellation of terms corresponding to conditioning on outside blocks when independence holds.

Linearity of expectation and combinatorial properties of weights guarantee that summation over subsets outside a block sums to one, allowing the reduction of sums to blockwise Shapley values.

This combination of linearity, independence, and symmetry underpins the blockwise decomposition of Pref-SHAP values.

C ALTERNATIVE APPROACHES: CAN OTHER VARIANTS OF SHAPLEY VALUE RESTORE THE BLOCK PATTERN?

A natural question is whether using other value functions can help Pref-SHAP respect the canonical block structure in cases where the original conditional value function leads to violation. Specifically:

- Off-manifold / Interventional / Marginal Shapley values Janzing et al. (2020) evaluate features outside the data distribution, which may help remove interdependencies across blocks. However, such approaches lack robustness Slack et al. (2020) because they evaluate the model on unrealistic samples, potentially making the explanations unreliable and vulnerable to adversarial manipulation.
- ManifoldSHAP Taufiq et al. (2023) attempts to stay on the data manifold by estimating it via kernel density estimators or score models. Since our setting involves estimating conditional expectations from a distributional perspective, Kernel Mean Embeddings could serve as a tool for implementing ManifoldSHAP in the Pref-SHAP context.
- Causal Shapley values Heskes et al. (2020) incorporate the underlying causal graph and account for structural dependencies between features. This allows attribution to reflect true causal contributions, thereby improving interpretability over interventional or marginal approaches in the presence of feature correlation.

As an illustration, consider the Gaussian setup:

$$X = \begin{bmatrix} X_a \\ X_b \end{bmatrix} = \begin{bmatrix} X_2 \\ X_4 \\ X_1 \\ X_3 \end{bmatrix},$$

where
$$X_a = \{X_2, X_4\}$$
, and $X_b = \{X_1, X_3\}$.

Assume $X \sim \mathcal{N}(\mu, \Sigma)$. The conditional expectation and variance of X_a given X_b are:

$$\mathbb{E}[X_a|X_b] = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b),$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}.$$

Notably, the conditional mean does not depend on Σ_{aa} . This implies that even if features within a block (e.g., X_a) are highly correlated, their internal dependency does not affect the conditional expectation as long as X_b is fixed. However, dependency *across* blocks will break the block decomposition property under the conditional value function.

Hence, while conditional value functions capture statistical dependence, they may cause Pref-SHAP to violate canonical structure, leading to unintuitive or biased attributions, especially when sensitive features are involved. In such cases, alternative value functions like causal or manifold-based ones may provide more reliable and fair explanations.

D STUDY OF PREF-SHAP PROPERTIES W.R.T THE CANONICAL FORM

D.1 EFFECT OF CONSTANT FEATURES ON PREF-SHAP

• Consider a skew-symmetric function with features $u = [u_1, u_2], v = [v_1, v_2],$ where each item is drawn from a bivariate Gaussian distribution with zero mean:

$$Z_1 = [X_1, Y_1]^{\top}, \quad Z_2 = [X_2, Y_2]^{\top}, \quad Z_1, Z_2 \sim \mathcal{N}(0, \Sigma).$$

From 9, as $\mathbb{E}[X_i] = \mathbb{E}[Y_i]$, $\forall i$, we find $\Phi_1 \neq \Phi_2$ in general. Even if $\mathbb{E}[X_i] = \mathbb{E}[X_j] \ \forall i, j$, Φ_1 and Φ_2 differ unless additional constraints hold.

762

763

764

765

766

768 769

775 776

777 778 779

780

781

782 783

784

785

786

787

788

789

791

793

794

796

798

799

800 801 802

803 804

805

806

807

808

809

If
$$u_2 = v_2 = \mathbb{E}[X_1]$$
, then $\Phi_2 = 0$, but $\Phi_1 = f(u, v)$.

- If $\mathbb{E}[X_1] = 0$, then $\Phi_1 = \Phi_2 = \frac{1}{2}(u_1 v_1)u_2$. Thus, the Shapley values are equal, and this equality is due to the symmetry axiom. The symmetry axiom states that two features should receive equal Shapley values if they contribute equally across all coalitions. For the two-feature case, if $v(\{1\}) = v(\{2\}) \Rightarrow \Phi_1 = \Phi_2$, even when one of the features is held constant.
- To explore this further, define:

$$v(\{1\}) = \frac{1}{2} \left(u_1 \mathbb{E}[Y_2 \mid Y_1 = v_1] - v_1 \mathbb{E}[X_2 \mid X_1 = u_1] \right),$$

$$v(\{2\}) = \frac{1}{2} \left(v_2 \mathbb{E}[X_1 \mid X_2 = u_2] - u_2 \mathbb{E}[Y_1 \mid Y_2 = v_2] \right).$$

If the features are independent, the conditional expectations reduce to marginals, and equality $v(\{1\}) = v(\{2\})$ implies:

$$(u_1 - v_1)\mathbb{E}[X_2] = (v_2 - u_2)\mathbb{E}[X_1] \tag{12}$$

If $\mathbb{E}[X_1] = \mathbb{E}[X_2] = 0$, then $\Phi_1 = \Phi_2$ always. But, if $\mathbb{E}[X_1] = \mathbb{E}[X_2] \neq 0$, then

$$\Phi_1 = \Phi_2 \implies u_1 - v_1 = v_2 - u_2
\implies u_1 + u_2 = v_1 + v_2$$
(13)

- This may appear counterintuitive: the constant feature may get equal attribution even though it does not vary. This is not an artifact of the conditional expectation but a consequence of the symmetry axiom. Even if a feature is constant or non-informative in terms of variation, Pref-SHAP may still assign it equal attribution simply due to how it appears in the function and due to symmetry, unless the distribution is shifted. This suggests, Pref-SHAP attributions are not just about feature importance in terms of variance or marginal effect, but also about how the feature interacts structurally in the model and in the coalitional expectations.
- The model function $f(u,v) = u^{\top} A v$ involves pairwise products of features (e.g., $u_1 v_2$ u_2v_1). Thus, the contribution of one feature depends on its interaction with the other. A constant feature interacting with a varying one may still result in a nonzero attribution.
- This motivates the use of Pairwise Interaction Shapley values Sundararajan et al. (2020), Fumagalli et al. (2024), which quantify pairwise contributions directly. Interaction Shapley methods can attribute the output more intuitively in models where the output depends primarily on feature interactions, as is the case in skew-symmetric functions. These interactions are model-driven, and statistical correlation (e.g., in the Gaussian setting) further modulates their effect when conditional value functions are used.

REAL-WORLD EXPERIMENT

This experiment is based on classification, so Kernel Logistic Regression (KLR) is used for modeling the pairwise preferences and the concept of residue modeling is not applicable here. The data is generated uniformly at random and the pairwise comparisons with labels are split into training (80%), validation (10%), and test (10%) sets and the experiments are averaged over 5 folds. Hyperparameter tuning is done similar to that of synthetic experiments. We have mapped the 25 features into 4 hidden features and applied the skew symmetric function separately on the 2 blocks generated from the mapped features.



Comparison of Feature Importance Methods

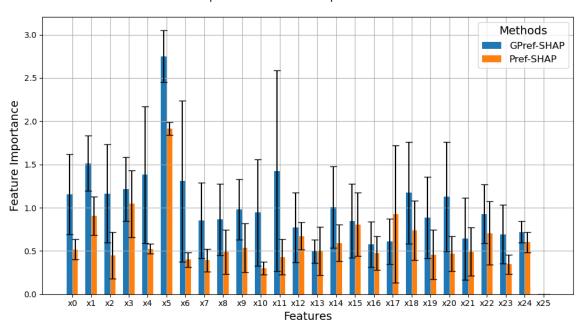


Figure 6: Pokemon dataset with 25 given features ['HP', 'Attack', 'Defense', 'Sp. Atk', 'Sp. Def', 'Speed', 'Legendary', 'Bug', 'Dark', 'Dragon', 'Electric', 'Fairy', 'Fighting', 'Fire', 'Flying', 'Ghost', 'Grass', 'Ground', 'Ice', 'Normal', 'Poison', 'Psychic', 'Rock', 'Steel', 'Water'], 'Speed' (x_5) is the most important feature in both methods.

ADDITIONAL SYNTHETIC DATA F

We also conduct experiments using synthetic data consisting of n = 100 items, where each item's features are sampled from a 4-dimensional Gaussian distribution with zero mean and a covariance matrix whose diagonal entries are 1 and off-diagonal entries are 0.7:

$$x^{(i)} \sim \mathcal{N}(0, \Sigma), \quad \text{for } i = 1, \dots, 100,$$

where $\Sigma_{ij} = 1$ and $\Sigma_{ik} = 0.7$ for $j \neq k$.

Figures 14, 15 and 16 are the synthetic experiments conducted on such highly correlated data. If we compare figures 16 and 17, we can see that when the features are highly correlated, the dummy/inactive feature x_3 gets more attribution than the case when the features are independent.

F.1 SANITY CHECK

Figures 17,18,19,20 are some experiments used for sanity check. In figure 17, the fourth/last feature is not used for generating mapped features and hence the skew-symmetric function. In figures 18, 19 and 20 items are generated using a 6-dimensional Gaussian distribution with mean zero and identity covariance: $x_i \sim \mathcal{N}(0, \mathbf{I})$ for $i = 1, \dots, 100$., but only the first 4 features are used for feature mapping and label generation, so the last two features act like dummy ones. We can observe from these plots that the dummy features get zero attribution in case of *Integrated Gradients* whereas Block Original and Generalized Pref-SHAP have nearly-zero or the lowest attribution for such features in all of them. But in figure 18, Pref-SHAP has more attribution for the dummy feature x_5 than x_0 which is an active feature in the function generation. Also, in figure 17, Pref-SHAP has more feature attribution for x_3 than that of x_2 .

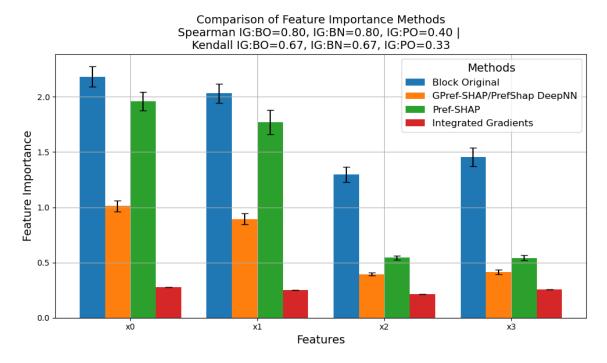


Figure 7: $x_0^2, x_1^2, x_2^2, x_3^2$

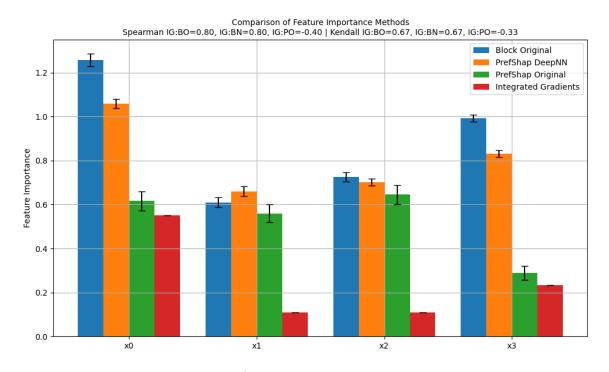


Figure 8: $x_0^2, x_1x_2, \sin(x_3), \cos(x_0)$

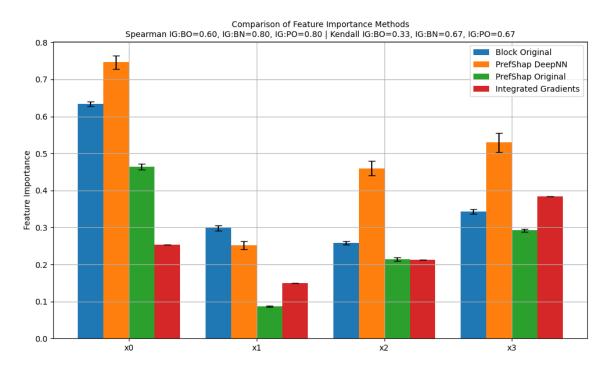


Figure 9: $sin(x_0), cos(x_1), sin(2x_2), cos(2x_3)$

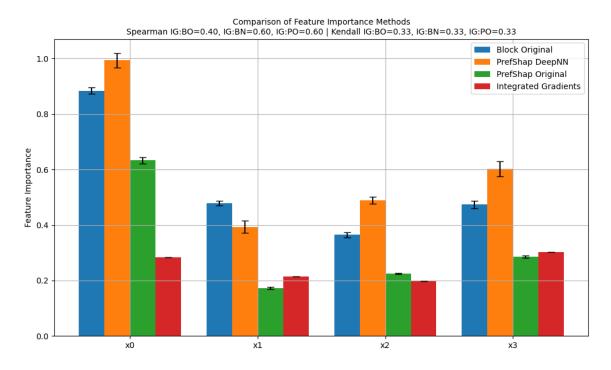


Figure 10: $sin(x_0)$, $cos(x_1)$, $sin(2x_2)$, $cos(2x_3)$, $sin(x_0 + x_1)$, $cos(x_2 - x_3)$

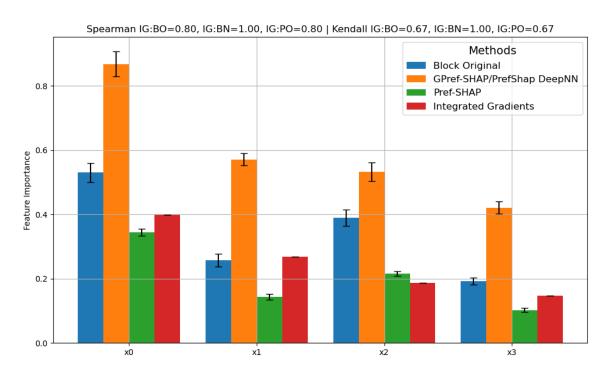


Figure 11: ReLU network(4 input features, 4 mapped features, 8 hidden layers, 16 nodes in each layer)

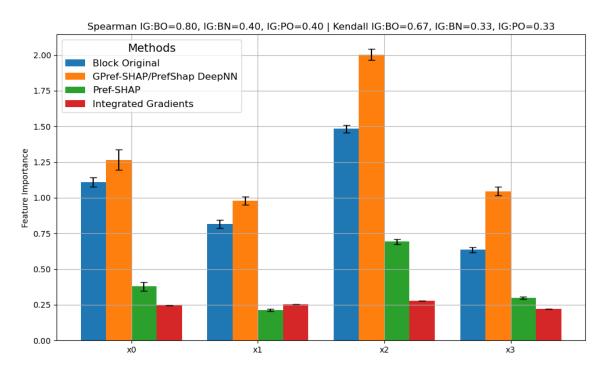


Figure 12: ReLU network(4 input features, 6 mapped features, 8 hidden layers, 8 nodes in each layer)

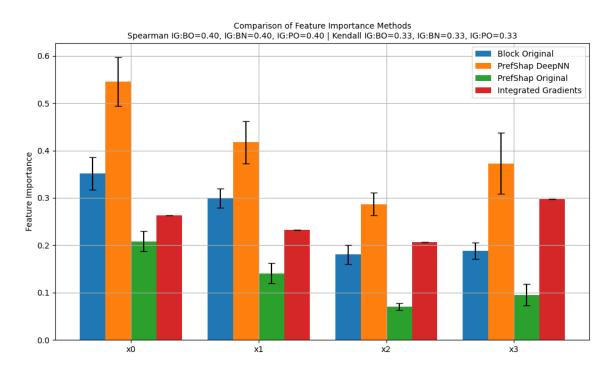


Figure 13: ReLU network(4 input features, 4 mapped features, 16 hidden layers, 16 nodes in each layer)

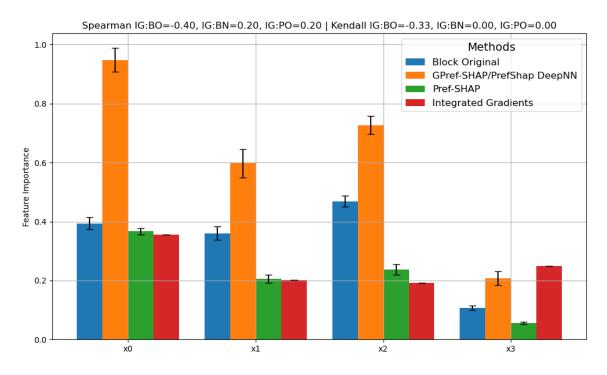


Figure 14: ReLU network(4 input features, 4 mapped features, 8 hidden layers, 16 nodes in each layer) for highly correlated data

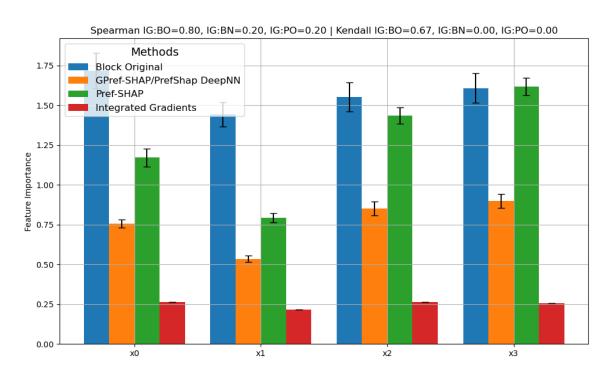


Figure 15: $x_0^2, x_1^2, x_2^2, x_3^2$ for highly correlated data

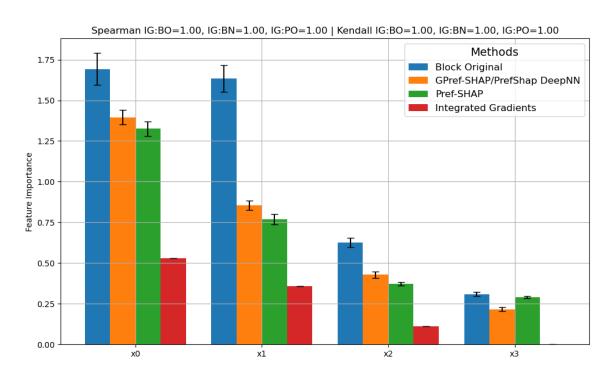


Figure 16: $x_0^2, x_1^2, x_0x_1, x_0x_2$ for highly correlated data

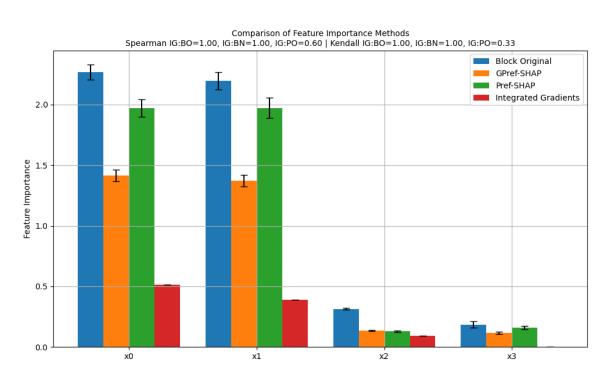


Figure 17: $x_0^2, x_1^2, x_0x_1, x_0x_2$ with x_3 as the inactive feature

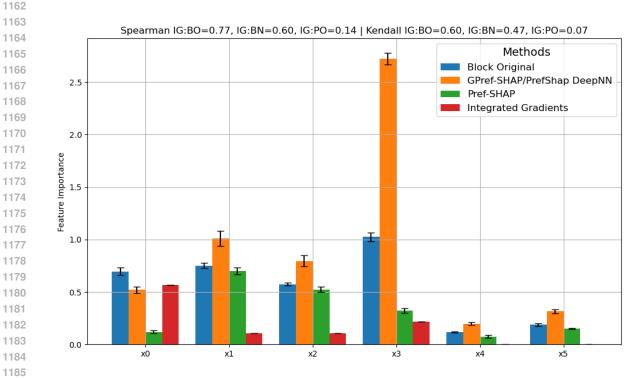


Figure 18: x_0^2 , x_1x_2 , $\sin(x_3)$, $\cos(x_0)$ with 2 inactive input features

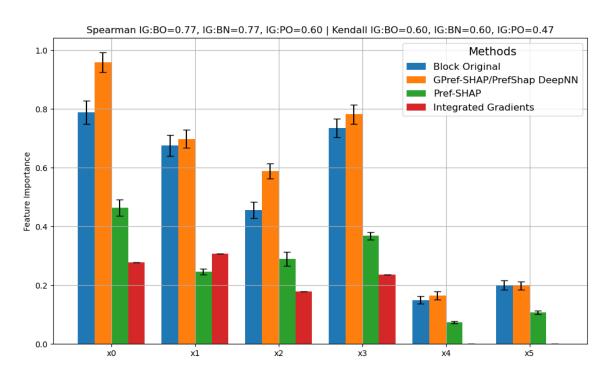


Figure 19: ReLU network(4 active + 2 inactive input features, 6 mapped features, 4 hidden layers, 16 nodes in each layer)

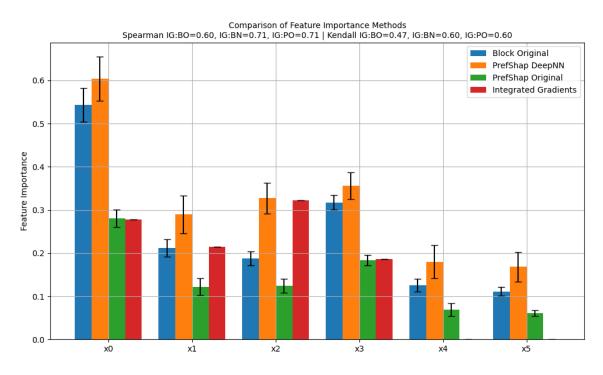
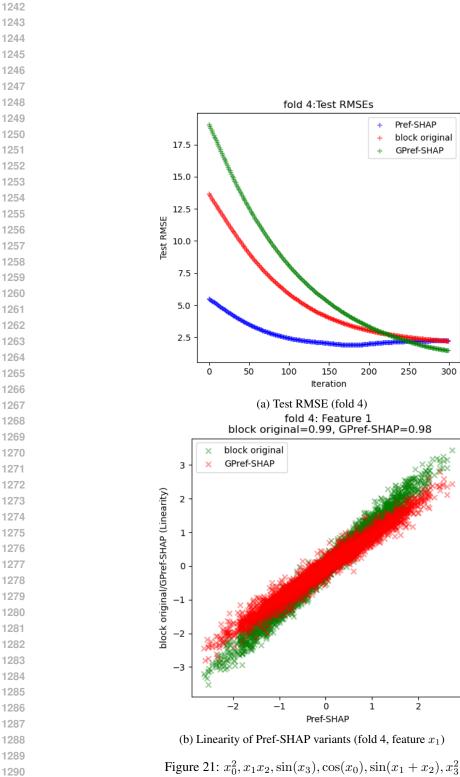


Figure 20: ReLU network(4 active + 2 inactive input features, 4 mapped features, 16 hidden layers, 16 nodes in each layer)



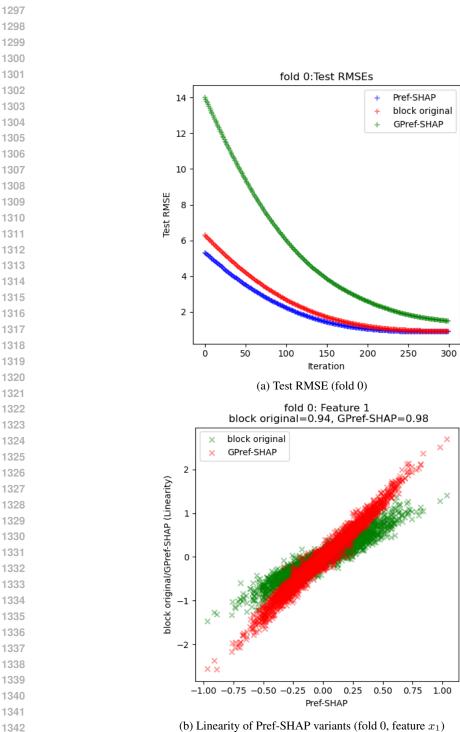


Figure 22: ReLU network(4 input features, 4 mapped features, 8 hidden layers, 16 nodes in each layer) with highly correlated features

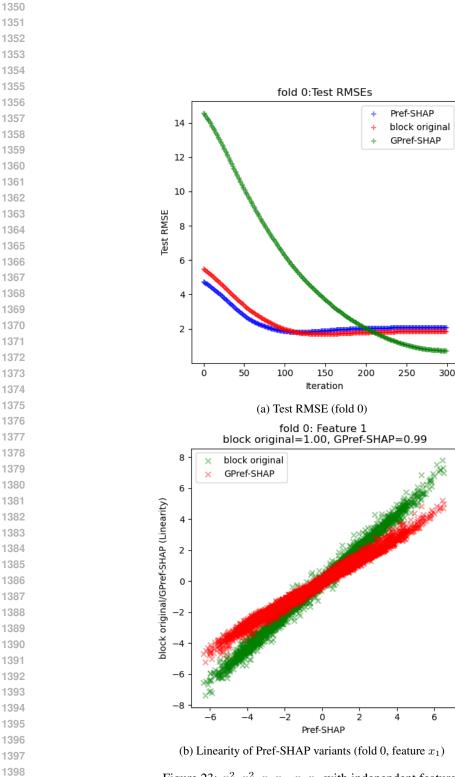


Figure 23: $x_0^2, x_1^2, x_0x_1, x_0x_2$ with independent features

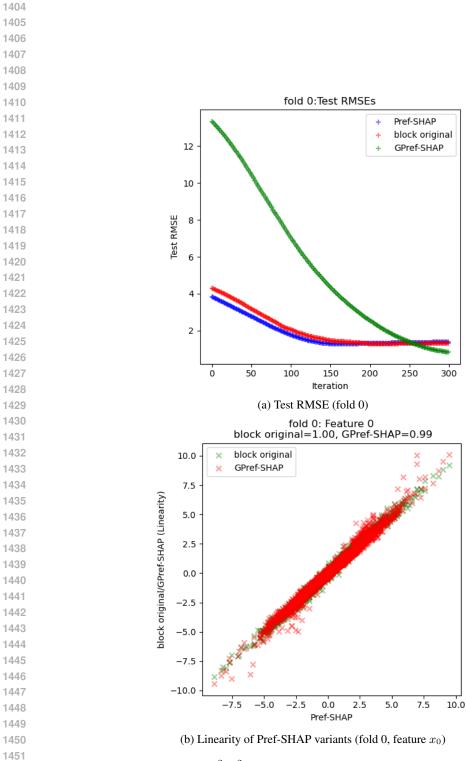


Figure 24: $x_0^2, x_1^2, x_0x_1, x_0x_2$ with highly correlated features