Unifying local and global model explanations by functional decomposition of low dimensional structures

Munir Hiabu
University of Copenhagen

Joseph T. Meyer
Heidelberg University

Marvin N. Wright
Leibniz Institute for Prevention
Research and Epidemiology – BIPS,
University of Bremen,
University of Copenhagen

Abstract

We consider a global representation of a regression or classification function by decomposing it into the sum of main and interaction components of arbitrary order. We propose a new identification constraint that allows for the extraction of interventional SHAP values and partial dependence plots, thereby unifying local and global explanations. With our proposed identification, a feature's partial dependence plot corresponds to the main effect term plus the intercept. The interventional SHAP value of feature k is a weighted sum of the main component and all interaction components that include k, with the weights given by the reciprocal of the component's dimension. This brings a new perspective to local explanations such as SHAP values which were previously motivated by game theory only. We show that the decomposition can be used to reduce direct and indirect bias by removing all components that include a protected feature. Lastly, we motivate a new measure of feature importance. In principle, our proposed functional decomposition can be applied to any machine learning model, but exact calculation is only feasible for low-dimensional structures or ensembles of those. We provide an algorithm and efficient implementation for gradient-boosted trees (xgboost) and random planted forest. Conducted experiments suggest that our method provides meaningful explanations and reveals interactions of higher orders. The proposed methods are implemented in an R package, available at `https://github.com/PlantedML/glex`.

1 INTRODUCTION

In the early years of machine learning interpretability research, the focus was mostly on single-value global feature importance methods that assign a single importance value to each feature. More recently, the attention has shifted towards local interpretability methods, which provide explanations for individual observations or predictions. Popular examples of the latter are LIME (Ribeiro et al., 2016) and SHAP (Shapley, 1953; Lundberg and Lee, 2017). The major reason for this shift is that local methods provide a more comprehensive picture than single-value global methods, most importantly in presence of nonlinear effects and interactions. This, however, neglects the fact that global methods can be more than single-value methods: Ideally, a global method provides useful information about the entire regression or classification function by providing an explanation for each feature and each interaction effect of arbitrary order, relative to the values they take on. As with local methods, this gives us an explanation for each observation. The crucial difference is that two observations which have a set of feature values in common receive the same explanation for main and interaction effects involving exclusively those features. We call a representation of a function with this property *global*. The components of a global explanation are not specific to all feature values of an observation but only to the feature values corresponding to the respective component. This does not only give a more comprehensive picture than local methods but the complete picture.

In summary, we distinguish between three properties of an explanation of a function.

- single-value global: Each feature $j \in \{1, \ldots, d\}$ receives a single descriptive value $v_j \in \mathbb{R}$ which does not depend on $x \in \mathbb{R}^d$.

- global: Each subset of features $S \subseteq \{1, \ldots, d\}$ receives a descriptive function $m_S : \mathbb{R}^S \to \mathbb{R}$ which only depends on values $x_S = \{x_k : k \in S\}$ and not on other values $x_{-S} = \{x_j : j \notin S\}$.

- local: Each subset of features $S \subseteq \{1, \ldots, d\}$ receives a descriptive function $\phi_S : \mathbb{R}^d \to \mathbb{R}$ which may depend on all values of $x \in \mathbb{R}^d$.

In this paper, we introduce a global explanation procedure by identifying components in a functional decomposition. We show that the proposed explanation is identical to q-interaction SHAP (Tsai et al., 2022), where q corresponds to the maximal order of interaction present in the model to be analyzed. Hence, we provide a new interpretation of SHAP values which is not game-theoretically motivated. Tsai et al. (2022) argue that it is practically not feasible to calculate q-interaction SHAP exactly because of computational complexity. However, the authors implicitly assume $q = d$, i.e., the highest order of interaction present in the initial estimator is equal to the number of features. We argue that this is not the case for many state-of-the-art machine learning algorithms that only fit low dimensional structures or ensembles of those. We exploit this fact and discuss an implementation that exactly calculates q-interaction SHAP for tree-based machine learning models. In principle, our results can be applied to any model and our algorithm can be applied to any tree-based model. However, since the number of components grows exponentially with increasing q, exact calculation is only feasible if q is sufficiently small. We provide a fast implementation for *xgboost* (Chen and Guestrin, 2016) and *random planted forest* (Hiabu et al., 2020).

As a result, one dimensional contributions m_k and two-dimensional contributions m_{jk} are one and two-dimensional real-valued functions that can be plotted. Furthermore, together with higher order contributions they can be used to decompose simple SHAP values into main effects and all involved interaction effects. Additionally, main and interaction components can be summarized into feature importance values. Beyond explaining feature effects, our proposed decomposition can be used to detect bias in models where LIME and SHAP fail (Slack et al., 2020) and reduce such bias by removing individual components from the decomposition.

The proposed methods are implemented in an R package, available at `https://github.com/PlantedML/glex`. Code to reproduce all figures and tables is available at `https://github.com/PlantedML/glex_paper`. Some further details and proofs to all lemmata and theorems are provided in the Appendix.

1.1 Motivating Example

We will give a toy example of how the interplay of correlations and interactions can give rise to misleading SHAP values. Consider the function $m(x_1, x_2) = x_1 + x_2 + 2x_1x_2$. The interventional SHAP value for the first feature is $\phi_1(x_1, x_2) = x_1 - E[X_1] + x_1x_2 - E[X_1X_2] + x_1E[X_2] - x_2E[X_1]$, see Appendix D. If the features are standardized,
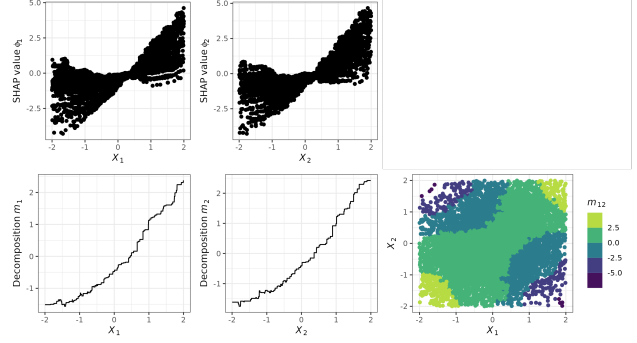

Figure 1: Simple example. Given an *xgboost* estimator $\hat{m}$ estimating the true function $m(x_1, x_2) = x_1 + x_2 + 2x_1x_2$, we calculate SHAP values (top row) and functional decomposition (bottom row).

i.e., X_1 and X_2 have mean zero and variance one, the expression reduces to

$$\phi_1(x_1, x_2) = x_1 + x_1x_2 - \mathrm{corr}(X_1, X_2).$$

Hence, e.g., if $\mathrm{corr}(X_1, X_2) = 0.3$, an individual with $x_1 = 1$ and $x_2 = -0.7$ would see a SHAP value of 0 for the first feature:

$$\phi_1(1, -0.7) = 0.$$

This is quite misleading, since clearly x_1 has an effect on the response m that is irrespective of the particular value of x_1. The underlying problem is that locally at $(x_1, x_2) = (1, -0.7)$, the main effect contribution and interaction contribution cancel each other out. Indeed, we will see that the SHAP value ϕ_1 can be decomposed into a main effect contribution of x_1, which is $m_1^*(x_1) = x_1 - 2\mathrm{corr}(X_1, X_2) = 0.4$ and an interaction contribution of $\{x_1, x_2\}$, which is $0.5 \times m_{12}^*(x_1, x_2) = x_1x_2 + \mathrm{corr}(X_1, X_2) = -0.4$. Figure 1 shows SHAP values and the functional decomposition of an *xgboost* model of the function $m(x_1, x_2)$. The SHAP values ϕ_1 and ϕ_2 contain main effect contributions m_1^* and m_2^* as well as half of the interaction contribution m_{12}^* each. The functional decomposition separates the contributions m_1^*, m_2^* and m_{12}^*.

Those familiar with SHAP values may argue that one can detect the non-zero impact of x_1 by plotting ϕ_1 over all instances (see Figure 1). This argument has two problems. Firstly, this does not change the misleading local value. Secondly, SHAP values can be quite arbitrary: If two estimators m and $\tilde{m}$ are equal on the support $\mathrm{supp}(X_1, X_2)$, the corresponding SHAP values at $x \in \mathrm{supp}(X_1, X_2)$ are generally not equal. This is because SHAP values are constructed by extrapolating outside the support of the data. Slack et al. (2020) has empirically demonstrated how this phenomenon can be exploited to hide the importance of protected features. One could ask for local explanations that do not extrapolate, hoping that this solves the problem. Unfortunately, this is not possible: If explanations are deduced only from the region with data support, those explanations are based on the

correlation structure of the features (Janzing et al., 2020). In particular a feature that has zero effect on the model output can still be assigned a value stemming from a correlated feature (Janzing et al., 2020; Sundararajan and Najmi, 2020). We conclude:

> *Local explanations that do not explicitly specify all interactions cannot lead to meaningful interpretations in the presence of correlated features.*

This is important to remember, noting that interpretation tools are usually used for black-box algorithms with the main purpose being to explain the model well in cases where interactions are present. Intuitively speaking:

> *A local interpretation that explicitly considers all interactions is a global interpretation.*

Hence the goal of this paper is to unify local and global explanations. We emphasize again that in contrast to simple SHAP or l-interaction SHAP ($l < q$), q-interaction SHAP provides a global explanation of a trained model.

1.2 Related Work

A functional decomposition for global interpretation of regression functions was introduced in the statistical literature in Stone (1994), and has been further discussed in Hooker (2007); Chastaing et al. (2012); Lengerich et al. (2020). These authors considered a different constraint called (generalized) functional ANOVA decomposition. In contrast, the constraint we introduce in this paper is linked to Shapley values. There is considerable literature on interactions and Shapley values. In cooperative game theory, pairwise player-player interactions were first considered by Owen (1972) and later generalized to higher-order interactions by Grabisch and Roubens (1999). In the machine learning context, arbitrariness of Shapley values due to interactions and correlations has been discussed in Kumar et al. (2020), Slack et al. (2020), Sundararajan and Najmi (2020), and possible solutions have been proposed in Zhang et al. (2021), Kumar et al. (2021), Harris et al. (2022), Ittner et al. (2021), Sundararajan et al. (2020). Recently, Tsai et al. (2022) introduced interaction SHAP for any given order and proposed an approximation to calculate them. In this paper, we introduce an identification constraint for a functional decomposition which connects to partial dependent plots (Friedman, 2001) and Shapley values with a value function that has recently been coined interventional SHAP (Chen et al., 2020). Alternative value functions have been discussed in Frye et al. (2020), Yeh et al. (2022). There are a variety of methods to obtain single-value global feature importance measures implied by SHAP. These include Casalicchio et al. (2018), Frye et al. (2020) and Williamson and Feng (2020), among others. Similar to our method suggested in Section 2.5,

these measures are weighted averages of local importance values. However, in contrast to our suggestion, most are motivated by additive importance measures (Covert et al., 2020).

Lastly, after the completion and upload of a first preprint of this paper, two related and relevant works were published. Bordt and von Luxburg (2022) show that for every SHAP-value function there exists a one-to-one correspondence between SHAP values and an identification in a functional decomposition. But they do not provide explicit solutions. Herren and Hahn (2022) describe an identification constraint that connects to observational SHAP. Our present paper finds an identification constraint that gives a one-to-one correspondence to interventional SHAP and provides a fast implementation for tree-based methods.

2 MAIN RESULTS

Let $(Y_i, X_{i,1}, \ldots, X_{i,d})$ be a data set of n i.i.d. observations with $X_{i,k} \in \mathbb{R}$, $i = 1, \ldots, n$; $k = 1, \ldots, d$. We consider the supervised learning setting

$$E[Y_i | X_i = x] = m(x),$$

where the function m is of interest and Y is a real valued random variable.[1] We assume that a reasonable estimator $\hat{m}$ of m has been provided.

2.1 Global Interpretation

With increasing dimension it can quickly get very hard, if not impossible, to visualize and thereby comprehend a multivariate function. Hence, a global interpretation of $\hat{m}$ is arguably only feasible if it is a composition of low-dimensional structures. Let us consider a specific decomposition of a multivariate function into a sum of main effects, bivariate interactions, etc., up to a d-variate interaction term.

$$
\begin{aligned}
\hat{m}(x) = \hat{m}_0 &+ \sum_{k=1}^{d} \hat{m}_k(x_k) \\
&+ \sum_{k<l} \hat{m}_{kl}(x_k, x_l) + \cdots + \hat{m}_{1,\ldots,d}(x) \\
&= \sum_{S \subseteq \{1,\ldots,d\}} \hat{m}_S(x_S).
\end{aligned}
\tag{1}
$$

The heuristic of the decomposition is that if the underlying function $m(x)$ only lives on low-dimensional structures, then m_S should be zero for most feature subsets S and the order of maximal interaction $q = \max\{|S| : m_S \neq 0\}$ should be much smaller than the number of features: $q << d$. This discussion, however, is not very meaningful before

---

[1] We use $Y_i \in \mathbb{R}$ for notational convenience. It is straightforward to extend to binary classification, whereas multiclass classification would require a slightly different procedure.

one has agreed on an identification; without suitable identification constraints, it is possible to change components on the right without altering the left hand side. We propose the following identification which we see as reasonable in its own right, but also connects interventional SHAP values (Chen et al., 2020; Janzing et al., 2020), partial dependence plots (Friedman, 2001) and de-biasing, as will be explained in the next three subsections.

Marginal identification: For every $S \subseteq \{1, \ldots, d\}$,

$$\sum_{T \cap S \neq \emptyset} \int \hat{m}_T(x_T) \hat{p}_S(x_S) \mathrm{d}x_S = 0, \qquad (2)$$

where $\hat{p}_S$ is some estimator of the density p_S of X_S.

Lemma 2.1. The marginal identification (2) is equivalent to

$$\sum_T \int \hat{m}_T(x_T) \hat{p}_S(x_S) \mathrm{d}x_S = \sum_{T \cap S = \emptyset} \hat{m}_T(x_T).$$

for every $S \subseteq \{1, \ldots, d\}$.

The next theorem states existence and uniqueness of a decomposition that satisfies the identification constraint (2) and describes the solution explicitly.

Theorem 2.2. Given any initial estimator $\hat{m}^{(0)} = \{\hat{m}_S^{(0)} | S \subseteq \{1, \ldots, d\}\}$, there exists exactly one set of functions $\hat{m}^* = \{\hat{m}_S^* | S \subseteq \{1, \ldots, d\}\}$ satisfying constraint (2) with $\hat{m}_{\mathrm{Sum}}(x) := \sum_S \hat{m}_S^{(0)}(x_S) = \sum_S \hat{m}_S^*(x_S)$. The functions are given by

$$\hat{m}_S^*(x_S) = \sum_{V \subseteq S} (-1)^{|S \setminus V|} \int \hat{m}_{\mathrm{Sum}}(x) \hat{p}_{-V}(x_{-V}) \mathrm{d}x_{-V}.$$

In particular $\hat{m}_S^*$ does not depend on the particular identification of $\hat{m}^{(0)}$.

Remark 2.3. Theorem 2.2, given Lemma 2.1, is a special case of a more general result in combinatorics where the solution is known as Möbius inverse (Rota, 1964). In Cooperative game theory the result is also known as Harsanyi dividend (Harsanyi, 1963).

The next corollary provides a practically more useful identification of the solution than Theorem 2.2.

Corollary 2.4. The solution $\hat{m}_S^*$ described in Theorem 2.2 can be re-written as

$$\hat{m}_S^*(x_S) = \sum_{T \supseteq S} \sum_{T \setminus S \subseteq U \subseteq T} (-1)^{|S| - |T \setminus U|} \qquad (3)$$

$$\times \int \hat{m}_T^{(0)}(x_T) \hat{p}_U(x_U) \mathrm{d}x_U.$$

Remark 2.5. In principle, Theorem 2.2 is a special case of Corollary 2.4, where we consider $\hat{m}_{\{1,\ldots,d\}}^{(0)} = \hat{m}_{\mathrm{Sum}}$

with $\hat{m}_S^{(0)} = 0$ for $S \neq \{1, \ldots, d\}$ as an initial estimator. Theorem 2.2 has a simpler notation. The following example shows the benefits of the Corollary: Assume $\hat{m}$ is generated by xgboost with 2 trees with depth 2. For simplicity, the first tree is split in dimensions $\{1, 2\}$, the second in $\{3, 4, 5\}$. We write $\hat{m}(x) = \hat{m}_{t_1}(x) + \hat{m}_{t_2}(x)$, where $\hat{m}_{t_i}$ comes from tree i. We can use Corollary 2.4 by choosing either (a) $\hat{m}_{1,\ldots,5}^{(0)}(x) = \hat{m}(x)$ or (b) $\hat{m}_{1,2}^{(0)}(x) = \hat{m}_{t_1}(x), \hat{m}_{3,4,5}^{(0)}(x) = \hat{m}_{t_2}(x)$. The resulting $\hat{m}^*$ is the same since it does not depend on the specific choice of $\hat{m}^{(0)}$ as long as $\hat{m}(x) = \sum_S m_S^{(0)}$. Choice (a) has 3 disadvantages compared to choice (b): First, with (a) we obtain $\hat{m}^*$ by calculating $2^5 - 1 = 31$ marginals. With (b) we only need to calculate $2^2 - 1 + 2^3 - 1 = 10$. Second, with (a) we must construct $\hat{p}_S$ with $|S|$ taking values up to 5. With (b) we only require $\hat{p}_S$ for $|S| \leq 3$. Third, with (a) we need to calculate up to 5-dimensional integrals. With (b), only up to 3-dimensional integrals must be calculated. For xgboost and random planted forest, the estimator $\hat{m}$ is the sum of low dimensional functions. For estimators without this property, calculating the entire decomposition is computationally infeasible for high dimensional input.

Example 2.6. Consider the setting of our simple example (Subsection 1.1), $m(x_1, x_2) = x_1 + x_2 + 2x_1x_2$, with X_1 and X_2 having mean zero and variance one. If $m(x) = m^*(x)$ and $m^*(x)$ satisfies the population version of the marginal identification (2), then

$$m^*(x_1, x_2) = m_0^* + m_1^*(x_1) + m_2^*(x_2) + m_{12}^*(x_1, x_2),$$

with

$$m_0^* = 2corr(X_1, X_2)$$
$$m_1^*(x_1) = x_1 - 2corr(X_1, X_2)$$
$$m_2^*(x_2) = x_2 - 2corr(X_1, X_2)$$
$$m_{12}^*(x_1, x_2) = 2x_1x_2 + 2corr(X_1, X_2).$$

2.2 Describing Interventional SHAP in Terms of our Global Explanation

We now show that there is a direct connection between the global explanation described in the previous section and interventional SHAP values. In particular, this connection describes interventional SHAP values uniquely without the use of game theoretically motivated Shapley axioms or a formula running through permutations, where the number of summands grows exponentially with d, see formula (7) in Appendix B. Fix a value $x_0 \in \mathbb{R}^d$. A local approximation at x_0 of the function $\hat{m}$ is given by

$$\hat{m}(x_0) = \phi_0 + \sum_{k=1}^d \phi_k(x_0), \qquad (4)$$

for constants $\phi_0, \phi_1(x_0), \ldots, \phi_d(x_0)$. Similar to the case of global explanations, the right hand-side is not identified.

Local explanations add constraints to equation (4) such that $\phi_k(x_0)$ is uniquely identified and best reflects the local contribution of feature k to $\hat{m}(x_0)$. Note that the explanation is not global because the explanation for feature k depends on the value of all features $x_0 = x_{0,1}, \ldots, x_{0,d}$ and not on $x_{0,k}$ only.

In what follows, we consider the identification leading to interventional SHAP values (Chen et al., 2020; Janzing et al., 2020), i.e., Shapley values with value function

$$v_{x_0}(S) = \int \hat{m}(x)\hat{p}_{-S}(x_{-S})dx_{-S}|_{x=x_0}. \quad (5)$$

See Appendix A for a definition of Shapley values.

Corollary 2.7. If $\hat{m}$ is decomposed such that (2) is fulfilled, then the interventional SHAP values are weighted averages of the corresponding components, where an interaction component is equally split to all involved features:

$$\phi_k(x) = \hat{m}_k^*(x_k) + \frac{1}{2}\sum_j \hat{m}_{kj}^*(x_{kj}) + \cdots + \frac{1}{d}\hat{m}_{1,\ldots,d}^*(x_{1,\ldots,d}).$$

Remark 2.8. A crucial point of Corollary 2.7 is that the local SHAP values can be described by the components of a global explanation. The result is also intriguing since usually the contribution or importance of a single feature in a general global representation as in (1) is a complicated interplay between various interactions, see Appendix D.

2.3 Describing Partial Dependence Plots in Terms of our Global Explanation

Given an estimator $\hat{m}$ and a target subset $S \subset \{1, \ldots, d\}$, the partial dependence plot (Friedman, 2001), ξ_S, is defined as

$$\xi_S(x_S) = \int \hat{m}(x)\hat{p}_{-S}(x_{-S})\mathrm{d}x_{-S}.$$

It is straight forward to verify that partial dependence plots are linked to a functional decomposition $\{\hat{m}_S^*\}$ identified via (2) through

$$\xi_S = \sum_{U \subseteq S} \hat{m}_U^*.$$

In particular if S is only one feature, i.e., $S = \{k\}$, we have

$$\xi_k(x_k) = \hat{m}_0^* + \hat{m}_k^*(x_k).$$

2.4 A De-Biasing Application Stemming from our Global Explanation

Assume U is a set of features that should not have an effect on $\hat{m}$. For example $U = \{\text{gender, ethnicity}\}$ in the case of non-discriminatory regulation requirements. Assume $\{1, \ldots, d\}$ is the disjoint union of U and V with a directed acyclic graph structure $X_U \to X_V \to m$, $X_U \to m$; as illustrated in Figure 2. Eliminating the causal relationship
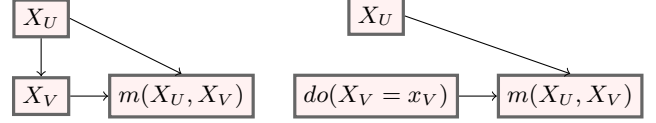

Figure 2: Left: Initial causal structure. Right: Causal structure after removing effect of X_U on X_V.

between X_V and X_U can be achieved via the do-operator, $do(X_V = x_V)$, that removes all edges going into X_V, see Figure 2. The function $E[m(X)|\, do(X_V = x_V))$ does not use information contained in X_U; neither directly nor indirectly; see also e.g. Kusner et al. (2017); Lindholm et al. (2022). Under the assumed causal structure, standard calculations (Pearl, 2009) lead to

$$E[m(X)|\, do(X_V = x_V)] = \int m(x)p_U(x_U)dx_U.$$

If $\hat{m}$ is identified via (2), then the de-biased version $\tilde{m}(x_{-U}) := \int \hat{m}(x)\hat{p}_U(x_U)dx_U$ can be extracted from $\hat{m}$ by dropping all components that include features in U:

$$\tilde{m}(x_{-U}) = \int \hat{m}(x)\hat{p}_U(x_U)dx_U = \sum_{S \subseteq V} \hat{m}_S(x_S). \quad (6)$$

2.5 Feature Importance

The global interpretation also provides a new perspective on feature importance. SHAP value feature importance for feature k is usually given by an empirical version of $E[|\phi_k(x)|]$. By Corollary 2.7,

$$E[|\phi_k(x)|] = E\left[\left|\sum_{S:k \in S} \frac{1}{|S|}\hat{m}_S^*(x_S)\right|\right].$$

In this definition, contributions from various interactions and main effects can cancel each other out, which may not be desirable. An alternative is to consider

$$E\left[\sum_{S:k \in S} \frac{1}{|S|}|\hat{m}_S^*(x_S)|,\right]$$

or to extend the definition of feature importance to interactions by defining feature importance as $E[|m_S^*(x_S)|]$, for a set $S \subseteq \{1, \ldots, d\}$.

Example 2.9. Going back to our simple example (Section 1.1), where $m(x_1, x_2) = x_1 + x_2 + 2x_1x_2$, SHAP feature importance for feature x_1 is an empirical version of

$$E\left[\left|X_1 - 2\mathrm{corr}(X_1, X_2) - \frac{1}{2}\{2X_1X_2 + 2\mathrm{corr}(X_1, X_2)\}\right|\right]$$
$$= E\left[|X_1 - X_1X_2 - \mathrm{corr}(X_1, X_2)\}|\right],$$

which merges main effect and interaction effect. Alternatively, one may consider

$$E\left[|X_1 - 2\mathrm{corr}(X_1, X_2)| + |X_1X_2 + \mathrm{corr}(X_1, X_2)\}|\right].$$

2.6 Remarks on our Main Results

The marginal identification (2) is new and the only identification that corresponds to partial dependence plots. Further, we are not aware of any other decompositions used in practice which allow for debiasing via post-hoc feature removal given the causal structure assumed in Subsection 2.4. Lastly, while the marginal identification corresponds to interventional SHAP, it is possible to construct other identifications that correspond to Shapley values with other value functions; e.g. a functional ANOVA constraint leads to observational SHAP (Herren and Hahn, 2022). To the best of our knowledge, we are the first to construct (and provide open-source) an algorithm which provides a global explanation for xgboost by including all higher-order interactions. In particular, all current implementations we are aware of only consider max. 2-way interactions (both in the case of interventional SHAP and observational SHAP).

3 EXPERIMENTS

We apply our method to several real and simulated datasets to show that the functional decomposition provides additional insights compared to SHAP values and SHAP interaction values. First, we show on real data that a global explanation can provide a more comprehensive picture than a local explanation method. Second, we show on real and simulated data that the same holds for the feature importance measure proposed in Section 2.5. Finally, we show that the functional decomposition allows post-hoc removal of features from a model, which can be used to reduce bias of prediction models. We performed all experiments with *xgboost* and *random planted forests*. The results with *xgboost* are presented in Sections 3.1-3.3 in the main paper, whereas the results with *random planted forests* are in Sections G.1-G.3 in the Appendix.

3.1 Global Explanations

As an example of a real data application, we apply our method to the *bike sharing* data (Fanaee-T and Gama, 2014), predicting the number of rented bicycles per day, given seasonal and weather information. Figure 3 shows SHAP values, main effects, 2-way interactions and 3-way interactions of the features *hour of the day* (hr, 0-24 full hours), *Temperature* (temp, normalized to 0-1) and *working day* (workingday, 0=no, 1=yes).

In the top row, we see that different SHAP values are observed for the same values of the features and conclude that SHAP values are not sufficient to describe the features' effects on the outcome, due to interactions. In the second row, the main effects from the decomposition show a strong effect of the hour of the day: Many bikes are rented in the typical commute times in the morning and afternoon. We also see a positive effect of the temperature and no main

effect of whether or not it is a working day. The 2-way interactions in the third row reveal strong interactions between the hour of the day and working day: On working days, more bikes are rented in the morning and less during the night and around noon. We also see that the temperature has a slightly higher effect on non working days and in the afternoon. In the bottom row, the 3-way interactions show that interactions between the hour of the day and the temperature are stronger on non working days than on working days.

We conclude that the full functional decomposition provides a more comprehensive picture of the features' effects, compared to usual SHAP value interpretations and 2-way interaction SHAP, as e.g. proposed by Lundberg et al. (2020). Note that, as described above, our methods do indeed provide the full picture, including all higher-order interactions, whereas Figure 3 only shows a subset of these interactions.

3.2 Feature Importance

As described in Section 2.5, the functional decomposition can also be used to calculate feature importance. Figure 4 shows the feature importance for the function $m(x) = x_1 + x_3 + x_2 x_3 - 2x_2 x_3 x_4$ and the bike sharing data from Section 3.1 based on SHAP values and our functional decomposition. For the simple function, the SHAP feature importance identifies x_1 and x_3 as equally important and x_2 and x_4 as less important but it gives no information about interactions. On the other hand, the feature importance based on the functional decomposition shows that x_1 has a strong main effect but no interactions, whereas x_2 and x_4 have only interaction effects but no main effects and x_3 both kinds of effects. Similarly, on the bike sharing data, the hour of the day (feature *hr*) and the temperature (*temp*) have both main and interaction effects, whereas the feature *working day* has 2-way interaction effects but no main effects (compare Figure 3). Note that both definitions of feature importance are based on absolute values of SHAP values or components m_S and thus are non-negative, in contrast to other methods of feature importance (Nembrini et al., 2018; Casalicchio et al., 2018).

3.3 Post-hoc Feature Removal

We show that our method can be used to remove features and all their effects, including interactions, from a model *post-hoc*, i.e., after model fitting. The idea is that in the setting of Subsection 2.4, using our decomposition, feature removal can be done by setting all components $m_S^* = 0$ for all $S \subseteq \{1, \ldots, d\}$ which include at least one component to be removed as in (6).

We trained models on simulated data and the *adult* dataset (Dua and Graff, 2017). Both models contained a feature *sex* or *gender*, which is a protected attribute and should not have an effect in fair prediction models (Barocas et al., 2019).
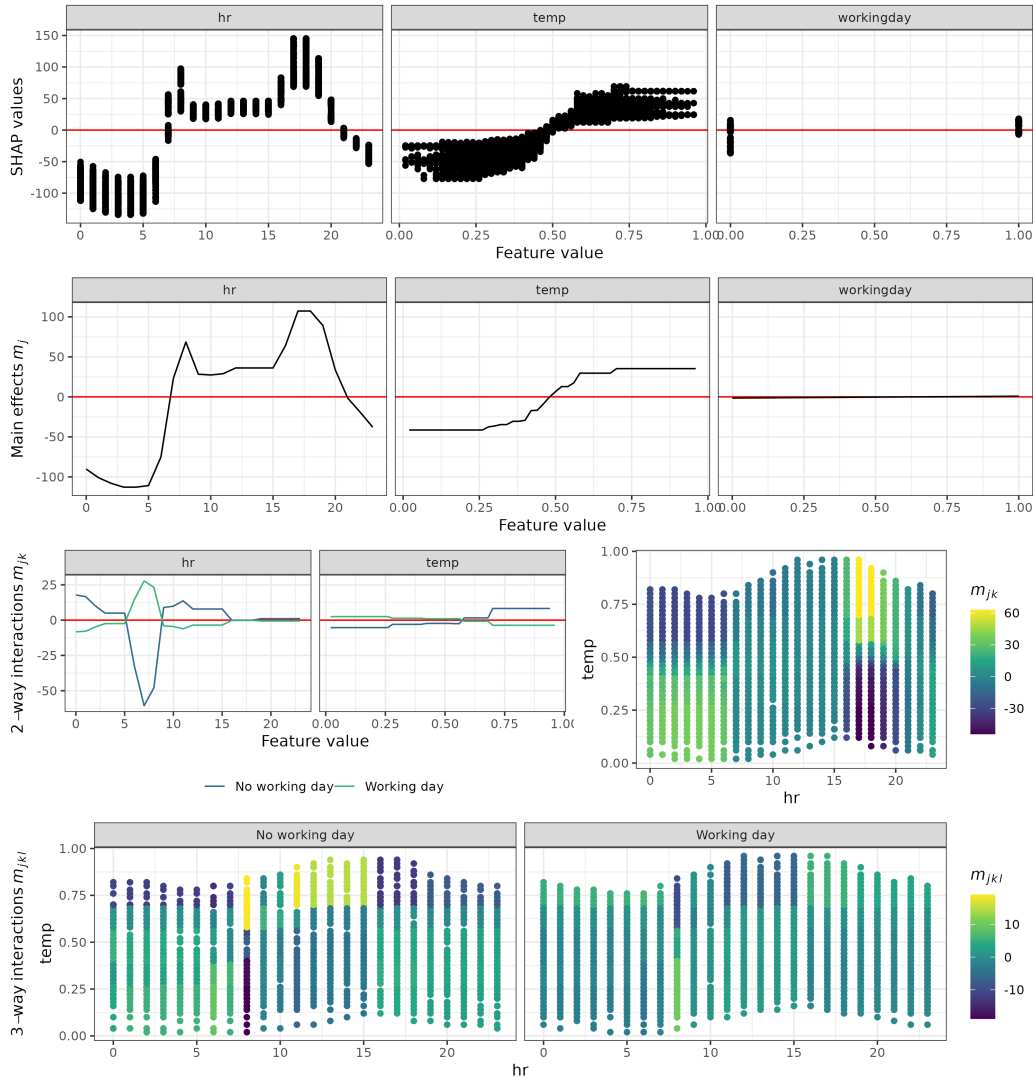
Figure 3: Bike sharing example (*xgboost*). SHAP values (top row), main effects (second row), 2-way interactions (third row) and 3-way interactions (bottom row) of the features *hour of the day* (hr, 0-24 full hours), *Temperature* (temp, normalized to 0-1) and *working day* (workingday, 0=no, 1=yes) of the bike sharing data.

In the simulation, we considered the simplified scenario where we predict a person's salary, based on their sex and weekly working hours. We set the weekly working hours to an average of 40 for men and to 30 for women. Salary was simulated as 1 unit (e.g. thousand Euro per year) per weekly working hour and an additional 20 for males (see Figure 2). Thus, men earn more for working longer hours (on average) and for being male per se. The first effect should be kept by a fair machine learning model, whereas the second effect is discriminating women. In the *adult* data, we have the same features *sex* and *hours* but we do not know the causal structure.

Figrue 5 shows the prediction for females and males of the full model, a refitted model without the protected feature *sex* and a decomposed model where the feature *sex* was removed

post-hoc. In the simulated data, we see that refitting the model does not change the predictions at all: Because of the high correlation between *sex* and *hours*, the effects of *sex* cannot be removed by not considering the feature in the model. Our decomposition, on the other hand, allows us to remove the (unwanted) direct effect of *sex* while keeping the (wanted) indirect effect through *hours*. On the *adult* data, we see a similar difference, but less pronounced.

4   CONCLUDING REMARKS AND LIMITATIONS

In this paper, we have introduced a way to turn local Shapley values into a global explanation using a functional decomposition. The explanation has a de-biasing application under
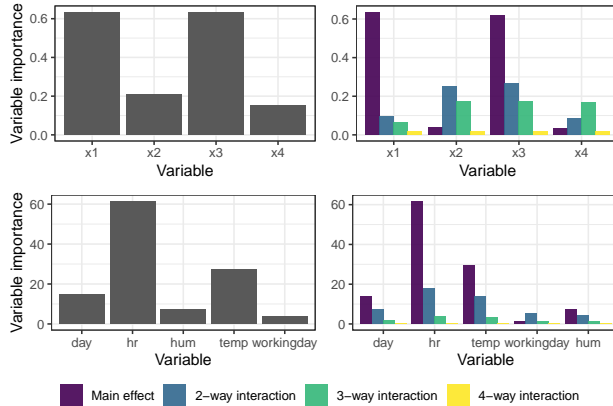
Figure 4: Feature importance (*xgboost*) for the function $m(x) = x_1 + x_3 + x_2x_3 - 2x_2x_3x_4$ (top row) and the bike sharing data from Section 3.1 (bottom row) based on SHAP values (left column) and our functional decomposition separately for main effects and interactions of different orders (right column).

the DAG structure given in Figure 2. This causal structure might be quite realistic in many fairness considerations, but the true causal structure is generally unknown. In this respect, it would be interesting to look into other causal structures that could motivate different identification constraints, which may connect to other local explanations than interventional SHAP. Indeed, Bordt and von Luxburg (2022) show that every SHAP-value function corresponds to a specific identification. Also, while our suggestions for feature importance measures paint a more precise picture in many cases, it is not directly motivated by a theoretical constraint, such as usual additive importance measures. It will require more research to back these ideas by theory. Another point not considered in this paper is the difference between the estimate $\hat{m}$ and a potential true function m. In particular, it is not clear if a method that estimates m well is also a good estimator for a selection of components m_S. This discussion is related to work done in double/debiased machine learning (Chernozhukov et al., 2018) and more general in semiparametric statistics, see e.g. Bickel et al. (1993). Moving forward, it could be interesting to modify out of the box machine learning algorithms to specifically learn the low dimensional structures well.

Ethical implications   Generally, explaining prediction models can help to reduce bias or discrimination. Specifically, our methods can be used to reveal higher-order interactions with protected attributes and by that detect bias and reduce such bias by post-hoc feature removal (see Section 3.3). However, there is more to *fair machine learning* than removing effects of protected attributes (see e.g. Barocas et al., 2019) and, as shown by Slack et al. (2020), machine learning explanation methods are not immune to (adversarial) attacks. Thus, results should be interpreted
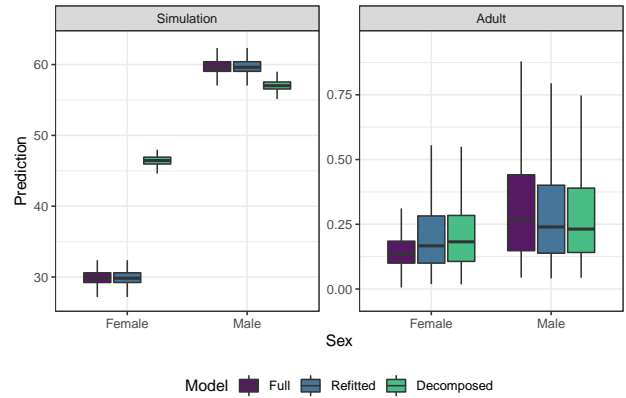
Setting	Median difference		
	Full	Refitted	Decomposed
Simulation	29.79	29.79	10.57
Adult	0.13	0.07	0.05

Figure 5: Post-hoc feature removal (*xgboost*). Predictions in a simulation (left) and the *adult* dataset for males and females of the full model, a refitted model without the protected feature *sex* and a decomposed model where the feature *sex* was removed post-hoc. The table below shows the median differences between females and males for the three models.

with care.

Acknowledgments

Bibliography

Barocas, S., Hardt, M., and Narayanan, A. (2019). *Fairness and Machine Learning*. http://www.fairmlbook.org.

Bickel, P. J., Klaassen, C. A., Bickel, P. J., Ritov, Y., Klaassen, J., Wellner, J. A., and Ritov, Y. (1993). *Efficient and adaptive estimation for semiparametric models*, volume 4. Springer.

Bordt, S. and von Luxburg, U. (2022). From Shapley values to generalized additive models and back. *arXiv preprint arXiv:2209.04012*.

Casalicchio, G., Molnar, C., and Bischl, B. (2018). Visualizing the feature importance for black box models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 655–670. Springer.

Chastaing, G., Gamboa, F., Prieur, C., et al. (2012). Generalized Hoeffding-Sobol decomposition for dependent variables-application to sensitivity analysis. *Electronic Journal of Statistics*, 6:2420–2448.

Chen, H., Janizek, J. D., Lundberg, S., and Lee, S.-I. (2020). True to the model or true to the data? *arXiv preprint arXiv:2006.16234*.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68.

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.

Dua, D. and Graff, C. (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Fanaee-T, H. and Gama, J. (2014). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2):113–127.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29:1189–1232.

Frye, C., Rowat, C., and Feige, I. (2020). Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability. *Advances in Neural Information Processing Systems*, 33:1229–1239.

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(4):547–565.

Harris, C., Pymar, R., and Rowat, C. (2022). Joint Shapley values: a measure of joint feature importance. In *International Conference on Learning Representations*.

Harsanyi, J. C. (1963). A simplified bargaining model for the n-person cooperative game. In *International Economic Review*, volume 4, pages 194–220.

Herren, A. and Hahn, P. R. (2022). Statistical aspects of SHAP: Functional ANOVA for model interpretation. *arXiv preprint arXiv:2208.09970*.

Hiabu, M., Mammen, E., and Meyer, J. T. (2020). Random planted forest: a directly interpretable tree ensemble. *arXiv preprint arXiv:2012.14563*.

Hooker, G. (2007). Generalized functional ANOVA diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3):709–732.

Ittner, J., Bolikowski, L., Hemker, K., and Kennedy, R. (2021). Feature synergy, redundancy, and independence in global model explanations using SHAP vector decomposition. *arXiv preprint arXiv:2107.12436*.

Janzing, D., Minorics, L., and Blöbaum, P. (2020). Feature relevance quantification in explainable AI: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pages 2907–2916. PMLR.

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.

Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., and Friedler, S. (2020). Problems with Shapley-value-based explanations as feature importance measures. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5491–5500. PMLR.

Kusner, M. J., Loftus, J., Russell, C., and Silva, R. (2017). Counterfactual fairness. *Advances in Neural Information Processing Systems*, 30.

Lengerich, B., Tan, S., Chang, C.-H., Hooker, G., and Caruana, R. (2020). Purifying interaction effects with the functional ANOVA: An efficient algorithm for recovering identifiable additive models. In *International Conference on Artificial Intelligence and Statistics*, pages 2402–2412. PMLR.

Lindholm, M., Richman, R., Tsanakas, A., and Wüthrich, M. V. (2022). Discrimination-free insurance pricing. *ASTIN Bulletin: The Journal of the IAA*, 52(1):55–89.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.

Nembrini, S., König, I. R., and Wright, M. N. (2018). The revival of the Gini importance? *Bioinformatics*, 34(21):3711–3718.

Owen, G. (1972). Multilinear extensions of games. *Management Science*, 18(5-part-2):64–79.

Pearl, J. (2009). *Causality*. Cambridge University Press.

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*, pages 1135–1144.

Rota, G.-C. (1964). On the foundations of combinatorial theory I. Theory of Möbius functions. *Zeitschrift*

für Wahrscheinlichkeitstheorie und Verwandte Gebiete, 2(4):340–368.

Shapley, L. S. (1953). A value for n-person games, contributions to the theory of games, 2, 307–317.

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. (2020). Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186.

Stone, C. J. (1994). The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics*, 22(1):118–171.

Sundararajan, M., Dhamdhere, K., and Agarwal, A. (2020). The Shapley Taylor interaction index. In *International Conference on Machine Learning*, pages 9259–9268. PMLR.

Sundararajan, M. and Najmi, A. (2020). The many Shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR.

Tsai, C.-P., Yeh, C.-K., and Ravikumar, P. (2022). FaithShap: The faithful Shapley interaction index. *arXiv preprint arXiv:2203.00870*.

Williamson, B. and Feng, J. (2020). Efficient nonparametric statistical inference on population feature importance using Shapley values. In *International Conference on Machine Learning*, pages 10282–10291. PMLR.

Yeh, C.-K., Lee, K.-Y., Liu, F., and Ravikumar, P. (2022). Threading the needle of on and off-manifold value functions for Shapley explanations. In *International Conference on Artificial Intelligence and Statistics*, pages 1485–1502. PMLR.

Zhang, H., Xie, Y., Zheng, L., Zhang, D., and Zhang, Q. (2021). Interpreting multivariate Shapley interactions in DNNs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10877–10886.

A   SHAPLEY VALUES

Consider a value function v_{x_0} that assigns a real value $v_{x_0}(S)$ to each subset $S \subseteq \{1, \ldots d\}$. Shapley axioms provide a unique solution under the four axioms efficiency, symmetry, dummy and additivity (Shapley, 1953), see Appendix B. Defining $\Delta_v(k, S) = v(S \cup k) - v(S)$, the Shapley values are

$$\phi_k = \frac{1}{d!} \sum_{\pi \in \Pi_d} \Delta_v \left(k, \{\pi(1), \ldots, \pi(k-1)\} \right) \tag{7}$$

$$= \frac{1}{d!} \sum_{S \subseteq \{1, \ldots, d\} \setminus \{k\}} |S|!(d - |S| - 1)! \Delta_v(k, S), \tag{8}$$

where Π_d is the set of permutations of $\{1, \ldots, d\}$. We follow Janzing et al. (2020) and define interventional SHAP values as Shapley values with the value function

$$v_{x_0}(S) = \int \hat{m}(x) \hat{p}_{-S}(x_{-S}) dx_{-S}|_{x=x_0}, \tag{9}$$

which is also the version implemented in TreeSHAP (Lundberg et al., 2020).

B   SHAPLEY AXIOMS

Given a function m, a point x_0, and a value function v, the Shapley axioms (Shapley, 1953) are

- **Efficiency**: $m(x_0) = \phi_0 + \sum_{k=1}^d \phi_k(x_0)$.

- **Symmetry**: Fix any $k, l \in \{1, \ldots, d\}, k \neq l$. If $v_{x_0}(S \cup k) = v_{x_0}(S \cup l)$, for all $S \subseteq \{1, \ldots d\} \setminus \{k, l\}$, then $\phi_k(x_0) = \phi_l(x_0)$

- **Dummy** If $v_{x_0}(S \cup k) = v_{x_0}(S)$, for all $S \subseteq \{1, \ldots d\} \setminus \{k\}$, then $\phi_k = 0$

- **Linearity** If $m(x_0) = m^1(x_0) + m^2(x_0)$, then $\phi_k(x_0) = \phi_k^1(x_0) + \phi_k^2(x_0)$, where ϕ^l is the explanation corresponding to the function m^l.

C   LEMMATA AND PROOFS

C.1   Lemmata

Lemma C.1 (Shapley (1953)). For every $U \subseteq \{1, \ldots, d\}$,

$$\int \hat{m}(x) p_{-U}(x_{-U}) dx_{-U} = \sum_{T \subseteq U} \hat{m}_T^*(x_T)$$

Proof.

$$\sum_{T \subseteq U} \hat{m}_T(x_T) = \sum_{T \subseteq U} \sum_{V \subseteq T} (-1)^{|T \setminus V|} \int \hat{m}_{\text{Sum}}(x) p_{-V}(x_{-V}) dx_{-V}$$

$$= \sum_{V \subset U} \int \hat{m}_{\text{Sum}}(x) p_{-V}(x_{-V}) dx_{-V} \sum_{S \subseteq \{1, \ldots, |U \setminus V|\}} (-1)^{|S|}$$

$$+ \int \hat{m}_{\text{Sum}}(x) p_{-U}(x_{-U})$$

$$= \int \hat{m}_{\text{Sum}}(x) p_{-U}(x_{-U}),$$

where the last equation follows from $\sum_{S \subseteq \{1, \ldots, |U-V|\}} (-1)^{|S|} = 0$, noting that a non-empty set has an equal number of subsets with an odd number of elements as subsets with an even number of elements. $\square$

C.2 Proofs

Proof of Lemma 2.1. Note that for every $S \subseteq \{1, \ldots, d\}$

$$\sum_T \int \hat{m}_T(x_T) p_S(x_S) \mathrm{d}x_S = \sum_{T \cap S = \emptyset} \hat{m}_T(x_T) + \sum_{T \cap S \neq \emptyset} \int \hat{m}_T(x_T) p_S(x_S) \mathrm{d}x_S.$$

Hence, condition (2) is equivalent to

$$\sum_T \int \hat{m}_T(x_T) p_S(x_S) \mathrm{d}x_S = \sum_{T \cap S = \emptyset} \hat{m}_T(x_T).$$

$\square$

Proof of Theorem 2.2. As stated in the main text, this result is a well known result. Here we provide an alternative proof by showing that the solution is the same as the solution of Corollary 2.4. In the proof of Corollary 2.4 we then show that the solution is the desired one.

We consider a fixed $S \subseteq \{1, \ldots, d\}$. We will make use of the fact that for a set $T \not\supseteq S$

$$\sum_{V \subseteq S} (-1)^{|S \setminus V|} \int \hat{m}_T^{(0)}(x_T) p_{-V}(x_{-V}) \mathrm{d}x_{-V} = 0, \tag{10}$$

and for a set $T \subseteq \{1, \ldots, d\}, T \supseteq S$

$$\{U : T \setminus S \subseteq U \subseteq T\} = \{T \setminus V : V \subseteq S\} \tag{11}$$

Combining (10)–(11), we get

$$\sum_{V \subseteq S} (-1)^{|S \setminus V|} \int \hat{m}_{\mathrm{Sum}}(x) p_{-V}(x_{-V}) \mathrm{d}x_{-V}$$

$$= \sum_{T \subseteq \{1, \ldots, d\}} \sum_{V \subseteq S} (-1)^{|S \setminus V|} \int \hat{m}_T^{(0)}(x_T) p_{-V}(x_{-V}) \mathrm{d}x_{-V}$$

$$= \sum_{T \supseteq S} \sum_{V \subseteq S} (-1)^{|S| - |V|} \int \hat{m}_T^{(0)}(x_T) p_{T \setminus V}(x_{-T \setminus V}) \mathrm{d}x_{T \setminus V}$$

$$= \sum_{T \supseteq S} \sum_{T \setminus S \subseteq U \subseteq T} (-1)^{|S| - |T \setminus U|} \int \hat{m}_T^{(0)}(x_T) p_U(x_U) \mathrm{d}x_U.$$

It is left to show (10)–(11). Equation (11) follows from straight forward calculations. To see 10, note

$$\sum_{V \subseteq S} (-1)^{|S \setminus V|} \int \hat{m}_T^{(0)}(x_T) p_{-V}(x_{-V}) \mathrm{d}x_{-V}$$

$$= \sum_{U \subseteq S \cap T} \sum_{W \subseteq S \setminus T} (-1)^{|S \setminus \{W \cup U\}|} \int \hat{m}_T^{(0)}(x_T) p_{-\{U \cup W\}}(x_{-\{U \cup W\}}) \mathrm{d}x_{-\{U \cup W\}}$$

$$= \sum_{U \subseteq S \cap T} \int \hat{m}_T^{(0)}(x_T) p_{-U}(x_{-U}) \mathrm{d}x_{-U} \sum_{W \subseteq S \setminus T} (-1)^{|S \setminus \{W \cup U\}|}$$

$$= \sum_{U \subseteq S \cap T} \int \hat{m}_T^{(0)}(x_T) p_{-U}(x_{-U}) \mathrm{d}x_{-U} \left(\sum_{W \subseteq S \setminus T, |W| = \mathrm{odd}} (-1)^{|S \setminus U| - 1} + \sum_{W \subseteq S \setminus T, |W| = \mathrm{even}} (-1)^{|S \setminus U|} \right)$$

$$= 0,$$

where the last equality follows from the fact that every non-empty set has an equal number of odd and even subsets. $\square$

Proof of Corollary 2.7. This proof is analogue to the proof of Lemma 3.1 in Shapley (1953). From C.1, We have $\int \hat{m}(x)p_{-U}(x_{-U})\mathrm{d}x_{-U} = \sum_{T \subseteq U} \hat{m}_T^*(x_T)$. Hence the game $\hat{m}$ with value function

$$v_{\hat{m}}(U) = \int \hat{m}(x)p_{-U}(x_{-U})\mathrm{d}x_{-U}$$

equals the game m^* with value function

$$v_{\hat{m}^*}(U) = \sum_{S \subseteq \{1,\ldots,d\}} \hat{m}_S^*(x_S)\delta_S(U), \quad \delta_S(U) = 1(S \subseteq U).$$

We now concentrate on the function $\hat{m}_S^*$ with value function $\hat{m}_S^*(x_S)\delta_S(U)$. We will show that for every non-empty $S \subseteq \{1,\ldots,d\}$,

$$\phi_k(x, \hat{m}_S^*(x_S)\delta_S(U)) = 1(k \in S)\,|S|^{-1}\,\hat{m}_S^*(x_S). \tag{12}$$

Here, $\phi_k(x, v)$ denotes the Shapley value for feature k at point x in a game with value function v. The proof is then completed by the linearity axiom, together with

$$\phi_k(x, \hat{m}_\emptyset^* \delta_\emptyset(U)) = \begin{cases} \hat{m}_\emptyset^*, & k = 0, \\ 0 & \text{else.} \end{cases}$$

The last statement follows from the dummy axiom. To show (12), let's assume that $j, k \in S$. Then for every $U \subseteq \{1,\ldots,d\}$, $\hat{m}_S^*(x_S)\delta_S(U \cup j) = \hat{m}_S^*(x_S)\delta_S(U \cup k)$, which, by the symmetry axiom, implies

$$\phi_j(x, \hat{m}_S^*(x_S)\delta_S(U)) = \phi_k(x, \hat{m}_S^*(x_S)\delta_S(U)).$$

Additionally, for $k \notin S$, we have $\phi_j(x, \hat{m}_S^*(x_S)\delta_S(U)) = 0$, by the dummy axiom. Hence we conclude (12) by applying (4). □

Proof of Corollary 2.4. Lemma C.1 implies that $\hat{m}^*$, as defined in (3), is a solution of the marginal identification. To see this, for $S \subseteq \{1,\ldots,d\}$, consider the following decomposition of $\int m^*(x)p_S(x_S)\mathrm{d}x_S$

$$\int \hat{m}^*(x)p_S(x_S)\mathrm{d}x_S = \sum_{T \cap S \neq \emptyset} \int \hat{m}_T^*(x_T)p_S(x_S)\mathrm{d}x_S + \sum_{T \cap S = \emptyset} \hat{m}_T^*(x_T).$$

Using Lemma C.1, we have

$$\int \hat{m}^*(x)p_S(x_S)\mathrm{d}x_S = \sum_{T \subseteq S^c} \hat{m}_T^*(x_T) = \sum_{T \cap S = \emptyset} \hat{m}_T^*(x_T),$$

which with the previous statement implies that $\hat{m}^*$ is a solution:

$$\sum_{T \cap S \neq \emptyset} \int \hat{m}_T^*(x_T)p_S(x_S)\mathrm{d}x_S = 0.$$

It is left to show that the solution is unique. Assume that there are two set of functions $\hat{m}^\circ$ and $\hat{m}^*$ that satisfy (2) with $\sum_S \hat{m}_S^\circ = \sum_S \hat{m}_S^*$. From Lemma 2.1, it follows that for all $S \subseteq \{1,\ldots,d\}$

$$\sum_{T \cap S = \emptyset} \hat{m}_T^\circ(x_T) = \sum_{T \cap S = \emptyset} \hat{m}_T^*(x_T),$$

implying $\hat{m}_T^\circ(x_T) = \hat{m}_T^*(x_T)$ for all $T \subseteq \{1,\ldots,d\}$. □

D   CONNECTING A GENERAL GLOBAL EXPANSION TO SHAP VALUES

If a regression or classification function m is not identified via (2), then calculating interventional SHAP values from such a decomposition leads to lengthy and non-trivial expressions. Here, we show how the terms up to dimension three in a general non-identified decomposition enter into a SHAP value. The following formula follows from straight forward calculations using (4).

For $v_x(S) = \int m(x)p_{-S}(x_{-S})dx_{-S}$, and $m(x) = \sum_{S \subseteq \{1,\dots,d\}} m_S(x_S)$, we get

$$
\begin{aligned}
\phi_1(x_0) = {}& m_1(x_1) - E[m_1(X_1)] \\
& + \frac{1}{2}\left\{ \sum_{j\neq1} m_{1j}(x_1,x_j) - E[m_{1j}(X_1,X_j)] \right. \\
& \qquad\quad \left. + \sum_{j\neq1} E[m_{1j}(x_1,X_j)] - E[m_{1j}(X_1,x_j)] \right\} \\
& + \frac{1}{3}\left\{ \sum_{j,k\neq1,j<k} m_{1jk}(x_1,x_j,x_k) - E[m_{1jk}(X_1,X_j,X_k)] \right. \\
& \qquad\quad + \sum_{j,k\neq1,j<k} E[m_{1jk}(x_1,X_j,X_k)] - E[m_{1jk}(X_1,x_j,x_k)] \\
& \qquad\quad + \frac{1}{2}\sum_{j,k\neq1,j<k} E[m_{1jk}(x_1,X_j,x_k)] - E[m_{1jk}(X_1,x_j,X_k)] \\
& \qquad\quad \left. + \frac{1}{2}\sum_{j,k\neq1,j<k} E[m_{1jk}(x_1,x_j,X_k)] - E[m_{1jk}(X_1,X_j,x_k)] \right\} \\
& + \frac{1}{4}\left\{ \sum_{j,k,l\neq1,j<k<l} m_{1jkl}(x_1,x_j,x_k,x_l) - E[m_{1jkl}(X_1,X_j,X_k,X_l)] \right. \\
& \qquad\quad + \sum_{j,k\neq1,j<k<l} E[m_{1jk}(x_1,X_j,X_k,X_l)] - E[m_{1jk}(X_1,x_j,x_k,x_l)] \\
& \qquad\quad + \frac{1}{2}\sum_{j,k\neq1,j<k<l} E[m_{1jk}(x_1,x_j,X_k,X_l)] - E[m_{1jk}(X_1,X_j,x_k,x_l)] \\
& \qquad\quad + \frac{1}{2}\sum_{j,k\neq1,j<k<l} E[m_{1jk}(x_1,X_j,x_k,X_l)] - E[m_{1jk}(X_1,x_j,X_k,x_l)] \\
& \qquad\quad + \frac{1}{2}\sum_{j,k\neq1,j<k<l} E[m_{1jk}(x_1,X_j,X_k,x_l)] - E[m_{1jk}(X_1,x_j,x_k,X_l)] \\
& \qquad\quad + \frac{1}{3}\sum_{j,k\neq1,j<k<l} E[m_{1jk}(x_1,x_j,x_k,X_l)] - E[m_{1jk}(X_1,X_j,X_k,x_l)] \\
& \qquad\quad + \frac{1}{3}\sum_{j,k\neq1,j<k<l} E[m_{1jk}(x_1,x_j,X_k,x_l)] - E[m_{1jk}(X_1,X_j,x_k,X_l)] \\
& \qquad\quad \left. + \frac{1}{3}\sum_{j,k\neq1,j<k<l} E[m_{1jk}(x_1,X_j,x_k,x_l)] - E[m_{1jk}(X_1,x_j,X_k,X_l)] \right\} \\
& \qquad\qquad \cdots
\end{aligned}
$$

E CALCULATING THE FUNCTIONAL DECOMPOSITION OF SHAP VALUES FROM LOW-DIMENSIONAL TREE STRUCTURES

Our proposed decomposition can be calculated from tree-based models by directly applying Corollary 2.4. Inspired by Lundberg et al. (2020), we first describe naïve algorithms for *xgboost* and *random planted forest* models and then describe an improved algorithm for *xgboost* that only needs a single recursion through each tree.

E.1 Naïve xgboost Algorithm

For all subsets of features $S \subseteq \{1, \ldots, d\}$, we calculate the decomposition $\hat{m}_S(x_i)$ for all observations of interest $x_i \in \mathbf{X}$ recursively for each tree with features T by considering all subsets U, T with $T \setminus S \subseteq U \subseteq T$. In each node of a tree, if the node is a leaf node we return it's prediction (e.g. the mean in CART-like trees). For internal (non-leaf) nodes, the procedure depends on whether the feature used for splitting in the node is in the subset U or not. If the feature is in the subset U, we continue in both the left and right children nodes, each weighted by the coverage, i.e. the proportion of training observations going left and right, respectively. If the feature is not in the subset U, we apply the splitting criterion of the node and continue with the respective node selected by the splitting procedure for observation x_i. See Algorithm 1 for the full algorithm in pseudo code.

Algorithm 1 NAÏVE XGBOOST ALGORITHM

  Procedure: DECOMPOSE($\mathbf{X}, \hat{m}(x)$)
  Input: Dataset to be explained $\mathbf{X} \in \mathbb{R}^{n \times d}$, tree-based model $\hat{m}(x)$ with B trees
  Output: Components $\hat{m}_S(x_i)$ for all $S \subseteq \{1, \ldots, d\}$ and $x_i \in \mathbf{X}$
  for $i \in 1, \ldots, n$ **do**
    for $S \subseteq \{1, \ldots, d\}$ **do**
      $\hat{m}_S(x_i) \leftarrow 0$
      for tree $\in \{1, \ldots, B\}$ **do**
        if $T \supseteq S$ **then**
          for $U : T \setminus S \subseteq U \subseteq T$ **do**
            $\hat{m}_S(x_i) \leftarrow \hat{m}_S(x_i) + (-1)^{|S| - |T \setminus U|} \text{RECURSE}(\text{tree}, U, x_i, 0)$
          end for
        end if
      end for
    end for
  end for
  Return: $\hat{m}_S$

  Procedure: RECURSE(tree, U, x_i, node)
  Input: Tree ID (tree), subset U, data point x_i, node ID (node)
  Output: Coverage-weighted prediction
  if ISLEAF(node) **then**
    Return: PREDICTION(node)
  else
    $j \leftarrow$ SPLIT-FEATURE(node)
    if $j \in U$ **then**
      $C_{\text{left}} \leftarrow$ COVERAGE(left-node)
      $C_{\text{right}} \leftarrow$ COVERAGE(right-node)
      Return: $C_{\text{left}} \text{RECURSE}(\text{tree}, U, x_i, \text{left-node}) + C_{\text{right}} \text{RECURSE}(tree, U, x_i, \text{right-node})$
    else
      if $x_i^j \leq$ SPLIT-VALUE(node) **then**
        Return: RECURSE(tree, U, x_i, left-node)
      else
        Return: RECURSE(tree, U, x_i, right-node)
      end if
    end if
  end if

E.2 Improved xgboost Algorithm

To improve the algorithm described in Section E.1 and Algorithm 1, we pre-calculate the contribution of each tree for all n observations and tree-subsets T in a single recursive procedure by filling an $n \times 2^D$ matrix, where D is the tree depth. In a second step, we just have to sum these contributions with the corresponding sign (see Corollary 2.4). See Algorithm 2 for the algorithm in pseudo code.

Algorithm 2 IMPROVED XGBOOST ALGORITHM

Procedure: DECOMPOSE($\mathbf{X}, \hat{m}(x)$)
Input: Dataset to be explained $\mathbf{X} \in \mathbb{R}^{n \times d}$, tree-based model $\hat{m}(x)$ with B trees
Output: Components $\hat{m}_S(x_i)$ for all $S \subseteq \{1, \ldots, d\}$ and $x_i \in \mathbf{X}$
for $S \subseteq \{1, \ldots, d\}$ **do**
  $\hat{m}_S(\mathbf{X}) \leftarrow 0$
end for
for tree $\in \{1, \ldots, B\}$ **do**
  $\mathcal{U} = \{U : U \subseteq T\}$
  $\hat{\mathbf{M}}(\mathbf{X}, \mathcal{U}) \leftarrow$ RECURSE(tree, $\mathcal{U}, \mathbf{X}, 0$)
  for $S \subseteq \{1, \ldots, d\}$ **do**
    if $T \supseteq S$ **then**
      for $U : T \setminus S \subseteq U \subseteq T$ **do**
        $\hat{m}_S(\mathbf{X}) \leftarrow \hat{m}_S(\mathbf{X}) + (-1)^{|S| - |T \setminus U|}\hat{\mathbf{M}}(\mathbf{X}, U)$
      end for
    end if
  end for
end for
Return: $\hat{m}_S$

Procedure: RECURSE(tree, $\mathcal{U}, \mathbf{X}$, node)
Input: Tree ID (tree), set of subsets $\mathcal{U}$, data matrix $\mathbf{X}$, node ID (node)
Output: Matrix of coverage-weighted predictions
if ISLEAF(node) **then**
  $\hat{\mathbf{M}}(\mathbf{X}, \mathcal{U}) \leftarrow$ PREDICTION($node$)
else
  $\hat{\mathbf{M}}_{\text{left}}(\mathbf{X}, \mathcal{U}) \leftarrow$ RECURSE(tree, $\mathcal{U}, \mathbf{X}$, left-node)
  $\hat{\mathbf{M}}_{\text{right}}(\mathbf{X}, \mathcal{U}) \leftarrow$ RECURSE(tree, $\mathcal{U}, \mathbf{X}$, right-node)
  $C_{\text{left}} \leftarrow$ COVERAGE(left-node)
  $C_{\text{right}} \leftarrow$ COVERAGE(right-node)
  $j \leftarrow$ SPLIT-FEATURE(node)
  $\mathbf{X}_{\text{left}} = (x_i : x_i^j \leq$ SPLIT-VALUE($node$))
  $\mathbf{X}_{\text{right}} = (x_i : x_i^j >$ SPLIT-VALUE($node$))
  for $U \in \mathcal{U}$ **do**
    if $j \in U$ **then**
      $\hat{\mathbf{M}}(\mathbf{X}, U) \leftarrow C_{\text{left}}\hat{\mathbf{M}}_{\text{left}}(\mathbf{X}, U) + C_{\text{right}}\hat{\mathbf{M}}_{\text{right}}(\mathbf{X}, U)$
    else
      $\hat{\mathbf{M}}(\mathbf{X}_{\text{left}}, U) \leftarrow \hat{\mathbf{M}}_{\text{left}}(\mathbf{X}_{\text{left}}, U)$
      $\hat{\mathbf{M}}(\mathbf{X}_{\text{right}}, U) \leftarrow \hat{\mathbf{M}}_{\text{right}}(\mathbf{X}_{\text{right}}, U)$
    end if
  end for
end if
Return: $\hat{\mathbf{M}}(\mathbf{X}, \mathcal{U})$

E.3   Random Planted Forest Algorithm

For the random planted forest (rpf) algorithm, we use a different approach. By slightly altering the representation of an rpf in Hiabu et al. (2020), the result of an rpf is given by a set

$$\hat{m}^{(0)} = \{\hat{m}^{(0)}_{S,b} | S \subseteq \{1, \ldots, d\}, \ b \in \{1, \ldots, B\}\},$$

where each estimator $\hat{m}^{(0)}_{S,b}$ can be represented by a finite partition defined by an $|S|$-dimensional grid (leaves) and corresponding values. Thus, we start with

- a grid $G_k = \{x_{k,1}, \ldots, x_{k,t_k}\}$ for each coordinate $k \in \{1, \ldots, d\}$,

- for each $S \subseteq \{1, \ldots, d\}, b \in \{1, \ldots, B\}$ an array representing the value of $m^{(0)}_{S,b}(x)$ for each coordinate $x \in \times_{k \in S} G_k$. Here x is considered to be the bottom left corner of a hyperrectangle.

Note that every tree-based algorithm can be described in such a manner. Given an estimator $\hat{p}_S$ and using this representation, directly calculating (3) is simple, where for each combination of sets $U, T \subseteq \{1, \ldots, d\}$ with $U \subseteq T$, we only need to calculate the term $\int \hat{m}^{(0)}_{T,b}(x_T)\hat{p}_U(x_U)\mathrm{d}x_U$ once and then add/subtract it to the correct estimators $\hat{m}^*_S$. See Algorithm 3 for the full algorithm in pseudo code.

Algorithm 3 NAÏVE RPF ALGORITHM

Procedure: DECOMPOSE($\mathbf{X}$, $\hat{m}(x)$)
Input: Dataset to be explained $\mathbf{X} \in \mathbb{R}^{n \times d}$, tree-based model with initial decomposition $\hat{m}^{(0)}_{S,b}(x)$ for $S \subseteq \{1, \ldots, d\}$ and trees $b = \{1, \ldots, B\}$, estimator $\hat{p}_S$
Output: Components $\hat{m}_S(x_i)$ for all $S \subseteq \{1, \ldots, d\}$ and $x_i \in \mathbf{X}$
for $i \in 1, \ldots, n$ **do**
  for $S \subseteq \{1, \ldots, d\}$ **do**
    $\hat{m}_S(x_i) \leftarrow 0$
  end for
  for $b \in \{1, \ldots, B\}$ **do**
    for $T \subseteq \{1, \ldots, d\}$ **do**
      for $U \subseteq T$ **do**
        $\mathrm{update}_{T,U} \leftarrow \int \hat{m}^{(0)}_{T,b}(x_{i,T\setminus U}, x_U)\hat{p}_U(x_U)\mathrm{d}x_U$
        for $S : T \setminus S \subseteq U, \ S \subseteq T$ **do**
          $\hat{m}_S(x_i) \leftarrow \hat{m}_S(x_i) + (-1)^{|S|-|T\setminus U|}\mathrm{update}_{T,U}$
        end for
      end for
    end for
  end for
  $\hat{m}_S(x_i) \leftarrow \hat{m}_S(x_i)/B$
end for
Return: $\hat{m}_S$

For the calculation of an estimator $\hat{p}_S$ in our simulations we used the following. For each $S \subseteq \{1, \ldots, d\}$, let $a_S(x)$ be the number of data points residing in the hyperrectangle with bottom left corner x for each coordinate $x \in \times_{k \in S} G_k$. For $|S|$-dimensional y we then set

$$\hat{p}_S(y) = \frac{a_S(x_y)}{\sum_{x \in \times_{k \in S} G_k} a_S(x)} \frac{1}{\mathrm{vol}(x)},$$

where x_y is the coordinate of the bottom left corner of the hyperrectangle which includes y and $\mathrm{vol}(x)$ is the volume of the hyperrectangle corresponding to x. Using this estimator, the updating function in the algorithm simplifies to

$$\mathrm{update}_{T,U} = \sum_{x_U \in \times_{k \in U} G_k} \hat{m}^{(0)}_{T,b}(x_{i,T\setminus U}, x_U)\hat{p}_U(x_U).$$
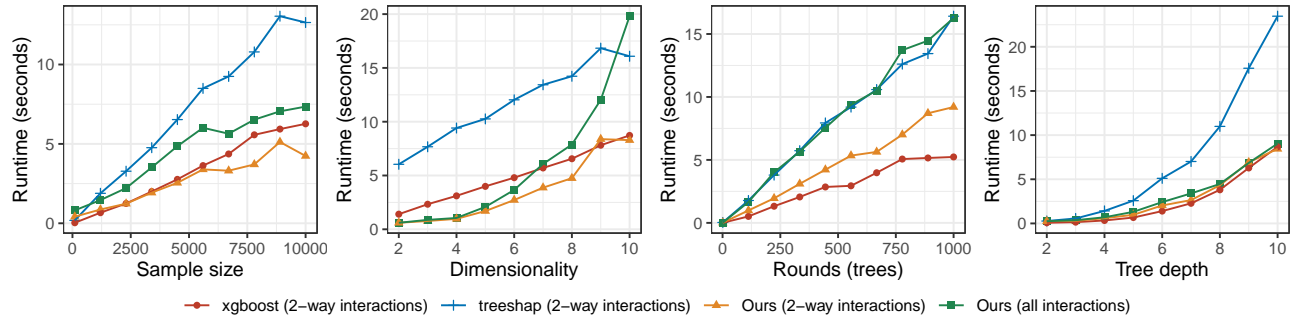
Figure 6: Runtime comparison. Runtime in seconds to calculate 2-interaction SHAP with different implementations, by sample size, dimensionality, number of rounds/trees and tree depth. In addition, runtime to calculate all interactions with our proposed method (green).

F COMPUTATIONAL COMPLEXITY

To address computational complexity to calculate all components of the decomposition, we performed a runtime comparison of our proposed algorithm with alternative methods to calculate TreeSHAP interactions. See Figure 6 for the results. We find that, to calculate 2-way interactions, our method is very competitive with the alternatives, i.e. our runtime is similar to that of the TreeSHAP implementation included in the xgboost library, which is also used by the official Python implementation of SHAP (https://github.com/slundberg/shap).

G EXPERIMENTS WITH RANDOM PLANTED FOREST

This section shows the simulation results when using the *random planted forest algorithm* as an estimation procedure. First of all, the results considering the motivating example from Section 1.1 are given in Figure 7. The following subsections include the results of experiments which where discussed in Section 3.

G.1 Global Explanations

Figure 8 includes the results discussed in Section 3.1 when considering the *random planted forest algorithm* instead of *xgboost*.

G.2 Feature Importance

Figure 9 includes the results discussed in Section 3.2 when considering the *random planted forest algorithm* instead of *xgboost*.

G.3 Post-hoc Feature Removal

Figure 10 includes the results discussed in Section 3.3 when considering the *random planted forest algorithm* instead of *xgboost*.

H COMPUTING ENVIRONMENT

A 64-bit Linux platform running Ubuntu 20.04 with an AMD Ryzen Threadripper 3960X (24 cores, 48 threads) CPU and 256 GByte RAM was used for all computations with R version 4.1.2.
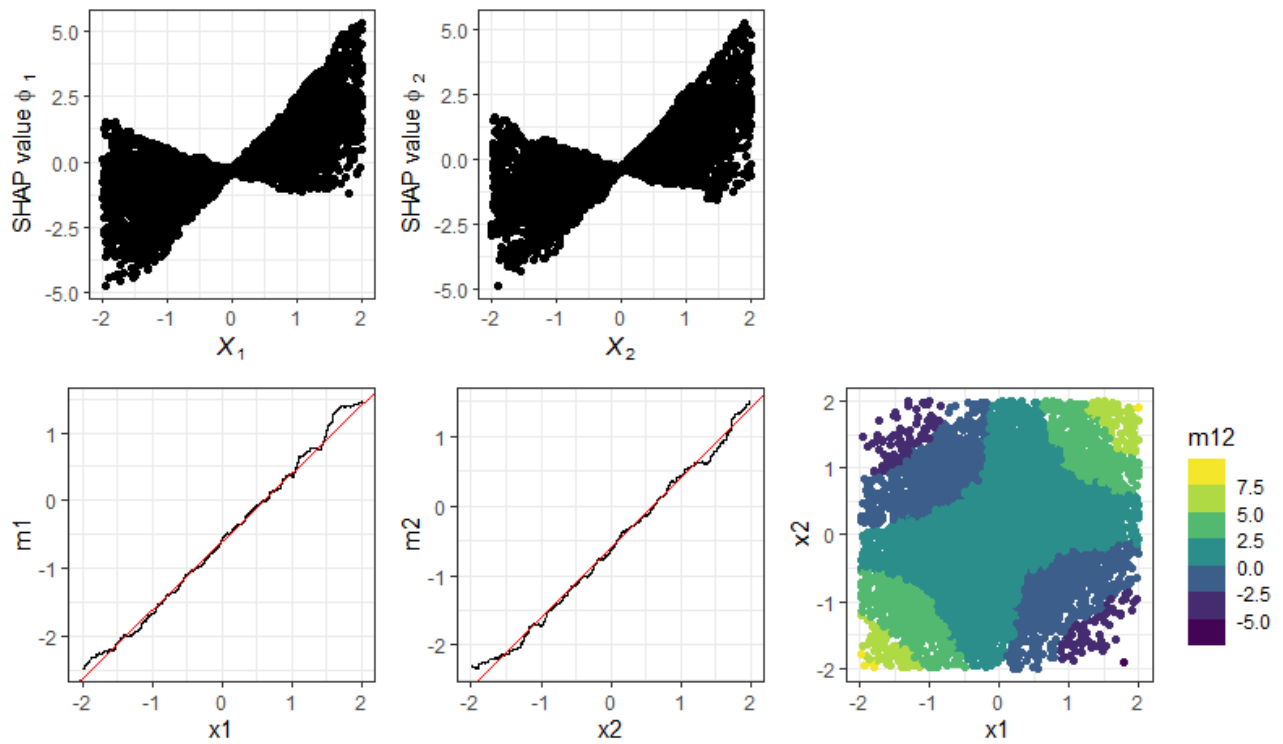
Figure 7: Simple example. SHAP values (top row) and functional decomposition (bottom row) of a *random planted forest* model of the function $m(x_1, x_2) = x_1 + x_2 + 2x_1x_2$. The red lines in the bottom row represent the SHAP values of the true function.
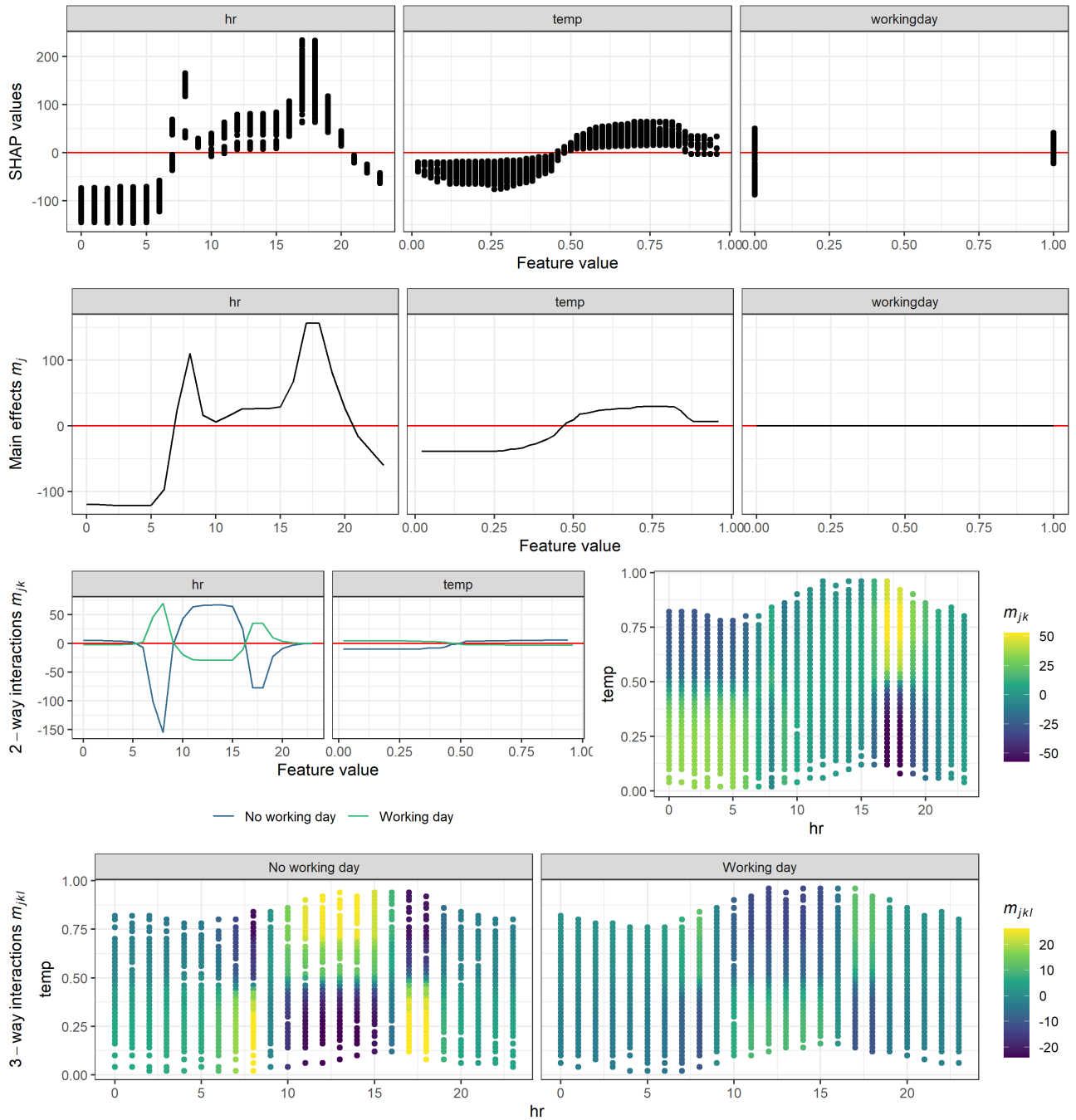
Figure 8: Bike sharing example (*random planted forest*). SHAP values (top row), main effects (second row), 2-way interactions (third row) and 3-way interactions (bottom row) of the features *hour of the day* (hr, 0-24 full hours), *Temperature* (temp, normalized to 0-1) and *working day* (workingday, 0=no, 1=yes) of the bike sharing data.

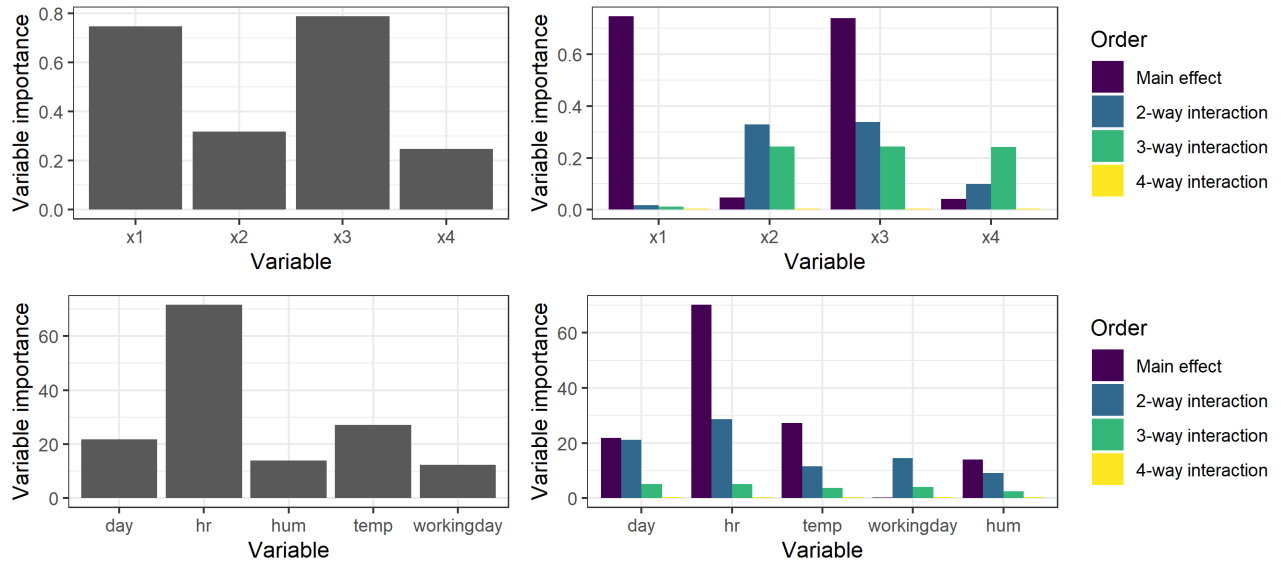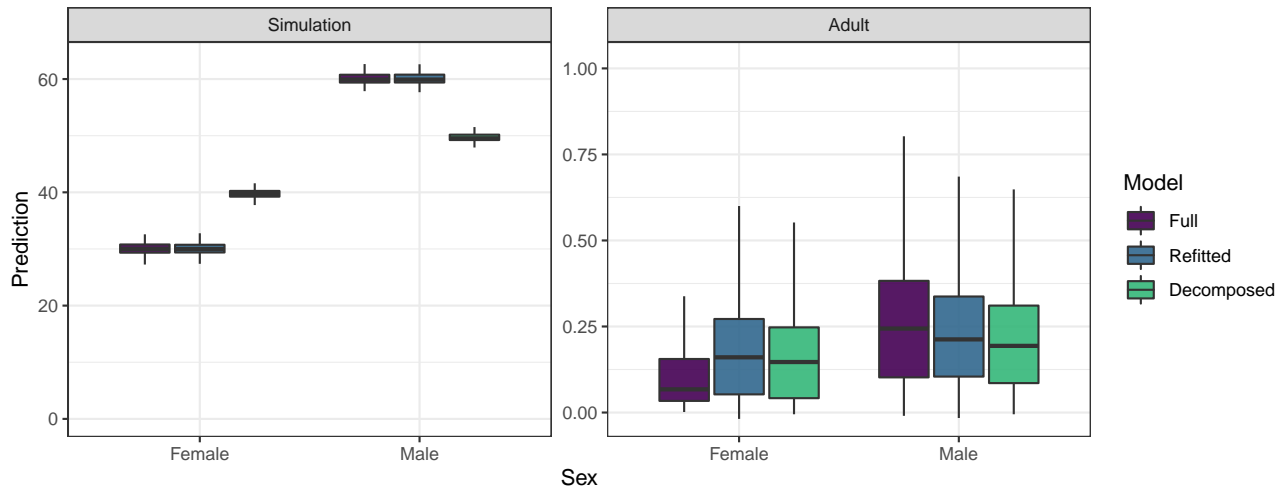Figure 9: Feature importance (*random planted forest*) for the function $m(x) = x_1 + x_3 + x_2 x_3 - 2 x_2 x_3 x_4$ (top row) and the bike sharing data from Section 3.1 (bottom row) based on SHAP values (left column) and our functional decomposition separately for main effects and interactions of different orders (right column).

| Setting | Median difference | | |
	Full	Refitted	Decomposed
Simulation	29.90	29.91	9.84
Adult	0.18	0.052	0.047

Figure 10: Post-hoc feature removal (*random planted forest*). Predictions in a simulation (left) and the *adult* dataset for males and females of the full model, a refitted model without the protected feature *sex* and a decomposed model where the feature *sex* was removed post-hoc. The table below shows the median differences between females and males for the three models.